Departamento de Electrónica e Informática - **Universidad Católica Nuestra Señora de la Asunción**

In Collaboration with

Department of Information Engineering and Computer Science - **University of Trento, Italy**

Final Project - Computer Science

Discovering and analyzing scientific communities using conference network

---

Author

Alejandro Mussi Campos Cervera

Supervisors

Prof. Dr. Fabio Casati

D.I.S.I. - University of Trento, Italy

Prof. Dr. Luca Cernuzzi

D.E.I. - Universidad Católica Nuestra Señora de la Asunción

March, 2010

Acknowledgements

I would like to thank my thesis advisor **Prof. Dr. Luca Cernuzzi**, who gave me the opportunity and confidence to develop this thesis in the context of a project coordinated by the University of Trento (UniTN). To **Prof. Dr. Fabio Casati**, my advisor from UniTN., who from the start showed his interest and support to make this work possible. His great vision and knowledge brought the necessary tools to meet the objectives.

To **Dr. Aliaksandr Birukou**, a postdoc researcher from UniTN., who guided me during the research and the documentation process of the thesis.

To the **University of Trento**, for granting me a scholarship and a warm work environment that made this work more enjoyable. Also, to the **Catholic University of Paraguay**, for the academic support it gave me and for the quick response in the procedures required for the scholarship award.

To my parents, **Luis** and **Susana**, who always support my projects, without them this would have not been possible. My brothers, **Tato** and **Diego**, faithful companions of my life. To my sisters **Carolina** and **Claudia**, who delight the family. All of them contribute in different ways to the achievment of my goals.

A special thanks to my sister **Carolina**, who helped me with the corrections of this book. Despite the short time available, she always finds the time and a way to help others.

My **grandparents**, **cousins** and **uncles**, who I have always received support from when I needed them.

A very special thanks to my girlfriend **Pati**, who has been always my trusted companion for all life endeavors, for her two visits while I was working in Italy giving me support and love.

To my **friends** and **colleagues** who make all this work much more enjoyable.

To my Family

Contents

Chapter 1

Goals and Motivations

1.1 Introduction

The increase number of scientific publications has made digital scientific literature search a difficult task and highly dependent of the researcher ability to search, filter and classify content. Most used scientific literature search engines and portals, such as Google Scholar [15], Citeseer [42] and ACM [3], use only simple text-base and citation-base score to rank the query result, and the rank is barely useful [38].

The number of references that a scientific publication has received (known as citations) determines the impact that the contribution has made to the community. Many methods (known as index) to measure or rank researchers are citation based [16, 12, 21]. A fair index for these is important because it is used to evaluate and compare researchers for different purpose, such as university recruitment, faculty advancement, award of grants, among others.

The world of science has many fields (Human, Social, Computer Science, etc.). Each field has different structures and publication dynamics. An example is the number of citations in the top-20 most cited journals in Computer Science is 4 times higher than the top-20 most cited journals in Social Science [43]. Therefore, it is unfair to compare researchers using citation-

based metrics without a context, in other words, the community they belong to. Different sizes of communities make currently most used metrics that measure the productivity or impact of researchers an unfair evaluation when comparing researchers from different communities since those with higher productivity are likely to produce more citations than communities with lower productivity.

The detection of scientific communities will allow us to improve two important activities in scientific research area. First, the *search* of scientific contributions. Being aware of the existing relations between scientific entities by knowing the communities they are part of, will enable more efficient search mechanisms since the domain of the queries can be narrowed down to particular communities, or can be sparse to different communities to obtain diversity of content. Moreover, having a framework that supports discovering scientific communities will provide the means for a better understanding of the social behavior in the scope of scientific research, enabling us the possibility to identify patterns in developments of projects, research trends, successful research profiles, and so on. Second, the *assessment* of people (researchers). In [1] is suggested that numerical indicators must not be used to compare researches or researchers across different disciplines. Since nowadays the boarders between disciplines are blurring, it is hard to define a priori the disciplines to which someone belongs. Ad-hoc and evolving communities can provide a better way for this.

Also, notice that communities may be hampered by construction, by how communities are constructed. Therefore, it is difficult to state a correct classification, especially because communities change over time, and people can belong to many communities in a particular period of time.

The detection of community structure on complex networks has become an interesting focus of investigation in different disciplines such as physic, social sciences, computer science, among others. Girvan and Newman were the first to introduce the property of community structure of a network [14], and an index to measure the quality of the structure called Modularity [33]. Many algorithms have been developed in the sake of detecting community structure of complex networks [47][33][7][31], but the vast majority of these algorithms do not take into account the overlapping of these communities [44].

This thesis presents an algorithm and a tool for discovering and evaluating scientific communities. The approach presented in this thesis combines different clustering algorithms for detecting overlapped scientific communities, based on conference publication data. The Community Engine Tool (CET) has implemented the algorithm and has been evaluated using the DBLP dataset, which contains information on more than 12 thousand conferences. The results showed that using our approach makes it possible to automatically produce community structure close to human-defined classification of conferences. The approach is part of a larger research effort aimed at studying how scientific communities are born, evolve, remain healthy or become unhealthy (e.g., self-referential), and eventually vanish.

1.2 Main Goal

The main goal of this thesis is to provide a model and a tool that support the detection and evaluation of scientific communities. Moreover, this thesis aims at proposing new metrics for the evaluation of individual productivity by normalizing it to the community.

1.3 Contribution Summary

The contribution of this work can be summarized as follow:

1. A model and a complete process for the detection of scientific communities using a conference network.
2. An algorithm for the detection of scientific communities.
3. New metrics for the evaluation of communities to improve the way scientific contents and authors are assesed.
4. An application that supports the detection and evaluation of scientific communities. The application is used on the Liquid Pub Project [22].

1.4 Outline

The work is organized as follow:

- **Chapter 2** presents the research line and scope of the thesis, introduction of the Liquid Pub project, which is the context of the development of this thesis, and related research done so far in order to better understand the actual state of the art of this thesis.
- **Chapter 3** describes the complete method we propose to achieve our goals. It starts by describing the problems we should face followed by the complete approach.
- In **Chapter 4**, the metrics for the evaluation of communities and people are presented.
- The details of the Community Engine Tool (CET) such as the architecture and implementation details are shown in **Chapter 5**.
- In **Chapter 6** are discussed the results of the experiments.
- Finally a **conclusion** of the work is given.

Chapter 2

Research Line and Scope

This Chapter describes the research line of this thesis: the LiquidPub Project, which is the actual context of the development of this work, and related research done in order to better understand the actual **state of the art** of this thesis.

The next sub-sections introduce the LiquidPub, a review of the theoretical concept of Scientific Community, followed by actual works and tools for the detection of scientific communities, and conclude with the current metrics used for the evaluation and impact of research and researchers.

2.1 Liquid Pub Project

The advent of the Web has made scientists improve the production/process in almost all areas. However, the world of scientific publication has been largely oblivious to the advent of the Web and to advances in Information and Communication Technologies (ICT). The way scientific knowledge is produced still follows the very same approach it did before the Web. The dissemination of scientific knowledge is still based on the traditional notion of "paper" publication and on peer review as the quality assessment method.

The mentioned problem was analyzed in the article **Publish and Perish:** ***why the current publication and review model is killing research and wasting your money*** [10]. It analyzed the actual model of dissemination of scientific contribution. They concluded that the traditional model is highly inefficient and has forgotten almost all the benefits of the Web.

The described scenario has motivated a deep analysis on the current way scientific knowledge is disseminated, and the Liquid Pub (LP) project is the outcome of this analysis, it aims to change the way scientific knowledge is created, evaluated, disseminated and consumed. The LP Project is a Framework Program 7 (FP7) and a funded research project in the Future and Emerging Technologies (FET). This thesis was made in the context of the LP Project and supported by a Grant of the University of Trento[1].

The goal of this project is to exploit novel technologies in order to enable a transition of the scientific paper from its traditional solid form, (i.e., a crystallization in space and time of a scientific knowledge artifact) to a Liquid Publication (or LiquidPub for short), that can take multiple shapes, evolves continuously in time, and is enriched by multiple sources. The intended benefits of this novel approach are:

- To increase the early circulation of innovative ideas, and hence foster a more effective dissemination.
- To optimize the time spent by researchers in creating, assessing and disseminating knowledge, while improving the quality of the paper selection processes for conferences and journals.
- To facilitate collaborative research efforts that builds upon previously developed knowledge.
- To develop a new way of credit attribution process based on social networks, team/-community work, collaborative problem solving, social reputation, and distribution of knowledge.

[1] http://www.dit.unitn.it/

- To deliver innovative services and products.

The LP project is coordinated by the University of Trento. The complete list of partners for the project are:

- **University of Trento** (Project coordinator). *Competences in knowledge management and assessment, software engineering, Web technologies.*
- **Springer Verlag**. *One of the leading companies in publishing scientific papers and books.*
- **Consejo Superior de Investigaciones Cientificas** (CSIC). *Competences in social networks, trust and reputation.*
- **Jean Nicod Institute** *Philosophers with competences in epistemology of IT.*
- **University of Fribourg** *Competences in modeling and analyzing competing behaviors of scientists.*

The LP project is currently in development phase and has defined a set of research areas that will help to achieve the goals. These research lines are:

- **Liquid Books**: The liquid book concept is a set of models and tools for writing and publishing books that are complete, up to date, of high quality, and targeted to specific group of readers. It is characterized by a set of authors that can share, reuse, modify and publish an edition of the book.
- **Liquid Journals**: the new frontiers of journals in the Web 2.0. Liquid Journals go beyond the traditional journal vision, proposing a new way of collecting, selecting and sharing scientific contributions with and within the Liquid Pub community.
- **Liquid Conference**: a platform for virtual meetings where invited people are presented for community discourse.

- **Research Evaluation**: analysis of how to assess the impact and the productivity of researchers in the Internet era, starting from how to improve traditional assessment process, like peer review, to new assessment methods exploiting the power of community.
- **Scientific Community**: the discovering and analysis of scientific communities could be useful in both the assessment and research process. Indeed, metrics to evaluate researchers (or contributions) can be normalized with respect to the community and then the impact of a researcher on one or more communities can be evaluated. Searching contributions through communities could also be useful to find interesting contents coming from different communities.
- **Scientific Knowledge Object (SKO)**: how to represent artifacts (e.g. documents, images, datasets) in the LiquidPub world, their evolution, reusability and composability.
- **Management Systems**: The LiquidPublications Management System (LPMS) is a tool that aims at providing a straight forward method for the specification and automated execution of processes concerned with the creation, dissemination, and evaluation of research work.
- **Services in the scientific publishing industry**: Oversee the evolution of the scientific publishing industry and gather information on how the LP project might affect it.
- **Licensing and copyright**: how to manage (and exploit) the various legal protections and freedoms related to scientific publishing, data and discourse.

This thesis is part of the *Scientific Community* research line of LP project[2]. This research line is focused on the analysis and detection of scientific communities with the purpose to improve the search and assessment of scientific content and researchers.

[2]http://project.liquidpub.org/research-areas/scientific-community

2.2 Scientific Community Concepts

A first step to this work is to define the type of community is seek, the concept of community in different disciplines, followed by the definition of the community that it aims to capture .

The definition of Scientific Communities can have different interpretations. In this field Kornfeld has made a complete analysis of the metaphor of Scientific Communities [20]. He defines a Scientific Community as a group of related scientists who interact with each other, and is often divided into sub-communities that work in particular areas.

Newman is one of the earliest to study scientific networks such as author collaboration network from different sources [29] [30] and has focus also on clustering them. He defines a community structure as a set of nodes densely connected within their cluster and less connected across other clusters/groups. In this definition, the composition of the communities is extended to any type of entity. In other words, an entity could represent a node in any type of network, it is not only applicable to social networks.

This work focuses on the new property for a graph that Newman proposed and named it Community Structure, and it is used in the context of Scientific World. Therefore, this thesis defines a Scientific Community as follows:

Definition 1. *A Scientific Community is defined as a set of scientists and any type of scientific entity, identified by a name, that are densely connected within the community and sparsely connected between other communities.*

A scientific entity is an abstract representation of all scientific content, such as journals, papers, conferences, among others. More details in Section 3.2.2. The Definition 1 aims at detecting communities of scientists (authors) and other scientific entities, such as conferences, and scientific publications, that are densely connected within the community and less densely connected among other communities.

The following section introduces current techniques and algorithms that will help the detec-

tion of scientific communities.

2.3 Community Detection Cluster Algorithms

In order to detect Scientific Communities, it is very important to apply clustering algorithms to identify groups of entities which are related. In the following sub-sections some concepts, and actual clustering and SNA techniques for the detection of communities are introduced.

2.3.1 Clustering Data

Clustering is a descriptive task that seeks to identify natural groupings in data. An important field of Knowledge discovery and data mining research focus on develop techniques to automatically discover such grouping [28].

In clustering analysis, it is well known that there is not a "better" algorithm. **Clustering algorithm** may be used in isolation to describe the data in a set of higher-level patterns, identifying groups of similar items based on their attribute values [46].

Traditional **graph partitioning algorithms** use the structure of a graph to find highly connected components. This approach focuses on the organization of nodes and edges in order to assign the nodes to a set of clusters in such a way that prescribed properties such as minimum cutsize or maximum connectivity are optimized.

A few clustering algorithms take into account both the attribute information and the structure of relationship in relational data. In [2] some open problems in relational clustering were discussed such as clustering heterogeneous data, and relation selection or extraction.

Both techniques, data clustering and graph partitioning, can be used to detect clusters on related data. In **Relational Cluster**, links confer a relationship between two objects in the same way that similar attributes values indicate a relationship.

	Running time $O(\cdot)$	Estimate k	Arbitrary shapes	Handle noise	One scan of data	Will stop
k-means	n					
k-medoids	n^2			•		
Agglo.	n^3					•
Divisive	n^2					•
EM Alg.	n					
Fract.	n				•	•
Refract.	n	•				
BIRCH	n			•	•	•
mrkd-EM	$n \cdot \log n$				•	
DBSCAN	$n \cdot \log n$	•	•	•	•	•
DENCLUE	n	•	•	•	•	•
DBCLASD	$n \cdot \log n$	•	•	•	•	•
STING	n	•	•	•	•	•
P. Filter	n	•			•	•
SOON	n^2	•				

Figure 2.1: Properties of Cluster Algorithms for large data set

Types of cluster-algorithms:

- **K-clustering**: partition the instances into k disjoin groups.
- **Hierarchical clustering**: produce a dendrogram of clusters, where the lowest level consists of a single instance.
- **Divisible algorithm** (top-down): it begins with the whole set and proceed to divide it into successively smaller clusters or simple instance.
- **Agglomerative algorithms** (bottom-up): it begins with each element as a separate cluster and merges them into successively larger clusters.

In order to apply clustering, it is necessary to define a measure to compute how close two objects are and it is commonly named as distance or similarity matrix. Any valid metric may be used as a similarity measure between pairs of observations.

In Relational Clustering, the input is a graph of related objects and the connection between two objects is defined as edges, not only by the similarity of their attributes. Graph Partitioning techniques were developed to use on graphs. The general goal is to partition the graph such as connections within cluster are maximized and connections between clusters are minimized.

Provide parameters to the cluster process makes the process "supervised". For example, K-means is a supervised algorithm because it needs the parameter k before computing the algorithm. A comparative table of most common cluster algorithms for large dataset is shown in 2.1.

2.3.2 Social Network Analysis

Social Network Analysis (SNA) has emerged as a key technique in modern sociology, and their metrics haven been used to analyze many different kinds of networks. SNA focus the attention more on the relationship between individuals rather than the attributes. Some metrics in SNA are:

- **Centrality**: this metric measures how well the node connect the network.
- **Degree**: the number of connections to others actors in the network.
- **Betweenness**: the degree of a node has between other nodes in the network.
- **Closeness**: the extend to which a node is close to all other nodes in the network.
- **Flow Betweenness Centrality**: the degree to which a node contributes to the sum of maximum flow between all pairs of nodes in the network.
- **Eigenvector centrality**: it measures the relevance of each node in the network.
- **Path Length**: measures the distance between two nodes in the network

Girvan and Newman have proposed a novel method for community detection cluster algorithm, built around the idea of using centrality indices to find community boundaries [14]. The process is based on cutting edges that have higher edge-betweeness in order to separates groups/communities. In each iteration computes the edge value of all edges in the graph $TimeComplex = O(mn)$ where $m = edges$ and $n = nodes$. So the algorithm has a time complexity of $O(mnt)$ where t =number of iterations. The betweenness value for all pair of vertex can be calculated in $O(mn)$ using breath-first search and a tree representation,

because calculating the shortest path for a pair of nodes take $O(n)$ and for all pairs $O(n2)$ [33].

The number of clusters is proportional to the number of iteration we compute. Therefore, a complete dendrogram is obtained when there is no more iterations to run (each instance represents a cluster). The new metric proposed by Girvan, called Edge-Betweenness, is an extension of Betweenness(of nodes) to Edges.

Some interesting points of the algorithm is that is simple and follows the definition of community on a graph which defines a community of a graph as a set of nodes densely connected within their cluster and less connected than across other clusters/groups, it provides a hierarchical output. Once the maximal iteration is computed, the navigation through different granularity is possible without computing the process again.

Some disadvantages of the algorithm are the time complexity in the worse case is $O(n^3)$, which is a problem when using large networks. Also, it does not provide an optimal cluster output.

Another interesting approach is proposed by Huberman, who proposes a community detection cluster algorithm with $TimeComplexity = O(N+E)$. It uses the same notion of community as Newman which define a community of a graph as a set of nodes highly connected within their cluster and less connected than across other clusters/groups.

The method avoids edge cutting and is based on the notion of voltage drops across networks [47]. The graph is represented as a electrical circuit where edges represent resistors, and two pole nodes represent the battery, then it applies Kirchoff equation to obtain the Voltage of each node. Then, the voltage will determine the community the node belongs to(community may have a voltage which characterizes it).

Good points about this algorithm is that the complexity on time scale linear respect to the size of the graph, and the community of a node can be computed without computing the complete graph. However, the algorithm does not compute an hierarchical cluster organization, this will not let the exploration of different granularities (dendrogram cut) on the cluster result.

Another weak point is that the number of communities/clusters should be given prior to the computation (supervised), and the batteries (pole nodes) should be computed in linear time and must be in different communities.

A metric called **Modularity** was introduced in [31] for the sake of measuring the property of community structure in a network (more details in Chapter 3). In other words, we can measure how good an algorithm has divided the network into communities. The modularity of the graph opens a wide range of heuristics based on modularity optimization [7].

The combination of community detection cluster algorithms and a index to measure the community structure will help us to evaluate the algorithm and select the best classification on a hierarchical output.

2.4 Metrics

The number of tournaments a golf player has won is a good well used metric to rank golf players, and definitely also for sponsors to support economically the player. In every discipline, metrics to rate people are necessary, and in Science is not the exception.

Researchers work hard to improve science by publishing their new contributions and therefore make our world a better place to live. But, how do we evaluate their work?. This section presents common metrics used to measure the impact of a scientist in order to understand how proposed metrics in this thesis can improve current evaluation methods by using the power of communities.

There are two important indexes that in conjunction can provide good references. One is the number of publications an author has made (known as publication number), and the other is the citation number. The citation number is defined as the number of references a contribution has received from another contribution to support part of its content. Citations have been widely used to measure the quality of the contribution, and many researchers have analyzed and concluded that despite the problems of using citation, such as self-citation, it

is still a good method to measure the scientific impact [5][35].

The time will tell us if a contribution has an important impact because the citations come after a long process of creation, evaluation and distribution of new publications, which reference a previous publication. One variation instead of using citation is proposed in [8] which uses the power of the Web and count the number of clicks/downloads of a paper in order to quickly evaluate the impact.

In this field, Hirsch [16] has proposed a metric called h-index, and is one of the most used metric till now to measure the productivity and impact of a scientist. It reflects both the number of publications (quantity) and the number of citations per publication (quality).

According to [16] a scientist has index h if h of his or her N_p papers have at least h citations each and the other $(Np - h)$ papers have h or less citations each. See Figure 2.2.

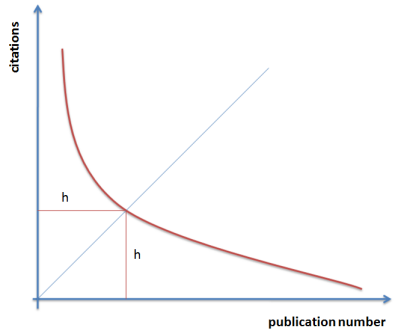

Figure 2.2: Schematic curve of number of citations versus publication number, with publications numbered in order of decreasing citations

Another interesting metric is the g-index, proposed by Egghe [12]. He states that the h-index is not sensitive to the level of the highly cited papers. The g-index is calculated as follows: *given a set of articles ranked in decreasing order of the number of citations that they received, the g-index is the (unique) largest number such that the top g articles received (together) at least* g^2 *citations.*

Besides the g-index, there are others citation based metrics that have emerged from the h-index, some of them are:

- **e-index**: proposed by Zhang [48]. The e-index aims at capturing the difference between scientist with similar h-index but different citation pattern. The e-index is defined as the (square root) of the surplus of citations in the h-set beyond h^2, i.e., beyond the theoretical minimum required to obtain a h-index of h.

- **Contemporary h-index**: proposed by Antonis Sidiropoulos et al. [40]. It aims at improving the h-index by giving more weight to recent articles, therefore rewarding academics who maintain a steady level of activity.

- **Age-weighted citation rate (AWCR) and AW-index**: proposed by Bihui Jin [18]. It measures the average number of citation of all contributions, adjusted for the age of each individual contribution.

- **Individual h-index**: It aims at reducing the effect of co-authorship by dividing the standard h-index by the average number of authors in the articles used to compute the h-index. The metric was inspired by Pablo Batista et al. [6].

- **Multi-authored h-index**: Michael Schreiber in his paper: *to share the fame in a fair way, h_m modifies h for multi-authored manuscripts* [39] proposes a simple modification of the h-index in order to take multiple co-authorship appropriately into account, by counting each paper only fractionally according to the number of authors.

Some useful tools to measure the scientific output using the described metrics are Harzings Publish or Perish [41], Google Scholar [15], and a new tool developed by the Liquid Pub project called Reseval [25].

The need to provide a context for current metrics has been analyzed by Batista [6], Mann [27] and Parra [36], and have motivated this thesis to use the power of the discovered communities in order to define context based metrics for the community and people, thus providing fairer metrics, especially when comparing authors from different disciplines.

Chapter 3

Unfolding Scientific Communities

In this section the complete process of the detection of scientific communities is described. We start from the description of the problems we need to face in order to achieve the goals, followed by the proposed model, the algorithm used for the detection of communities, and the creation of the community network.

3.1 Discovering Scientific Communities: Problem and Scope

The problem of modeling, managing and analyzing scientific communities, is presented to us as a wide range of different aspects that needs to be confronted.

The Figure 3.1 provides the list of problems to be confronted. Each of these problems has its own complexity and challenges.

- **Problem 1** - Scientific Data Extraction: the first step of the process is to provide the data for the framework. This problem is focused on extraction of data from different sources.
- **Problem 2** - Data Representation: the way of representing connections between entities

Figure 3.1: Scientific Communities problem stack

will define the shape of the communities. This issue is about establishing a model for communities and the data to be extracted for detection of them.

- **Problem 3** - Community Discovery Algorithm: the problem is about developing algorithms capable of detecting community structure.
- **Problem 4** - Naming: once communities are detected, each of them should be identified by a name that characterizes the community.
- **Problem 5** - Metrics: this problem is about proposing new community based metrics for the sake of improving the evaluation of scientific content and researchers.
- **Problem 6** - Community Mining: once communities are available, a problem is how to mine all of this data finding patterns and hidden information helping people to understand research activity and trends better.
- **Problem 7** - Evolution and Trends: as scientific communities are not static, methods to manage the evolution of communities along the time are necessary. This problem deals with the problem of designing and implementing business logic to support evolution of communities.
- **Problem 8** - Browsing: once information of communities is available, methods to query and navigate through this information are necessary. Thus, design and implementation a browsing interface for communities it is also an important problem in the scientific communities stack of problems.

- **Problem 9** - Visualization: this problem is about designing and implementing a visual model for communities that enable users to interact with communities.
- **Problem 10**- Reporting: reporting problem consists in developing tools, required for extracting, summarizing and reporting information about communities.

With the list of problems is intended to have a wide view of the different aspect to be confronted. However, some problems have been addressed deeply, and others have been only introduced, while addressing them is part of the future work.

3.2 Data Extraction and Representation

3.2.1 The Datasets

Different sources are available in different formats and ways to access. The Resman project [26], which is an ongoing project of LiquidPub, proposes a uniform way to access heterogeneous sources. Although, this thesis does not focus on proposing a better way to accessing different sources, it required when using the Internet as the dataset for discovering communities.

The data set used in this thesis is a DBLP dump, which is publicly bibliographic data source in XML format available at http://dblp.uni-trier.de/xml/. Most common bibliographic data source such as Citeseer, Google Scholar, and DBLP have name disambiguation problems, this may include that an author has multiple names, and multiple authors have the same name. According to [24], DBLP bibliographic information is maintained by massive human effort in order improve name consistency. We parsed the XML file and store it into a Relational Database for easy access and retrieval. For this case, a sub-set of the dblp dump is used, which contain all proceedings (12.227), all in-proceedings (747.752), and all authors (533.334) as of 08/03/2009.

The DBLP dump does not contain citation data for in-proceedings. Therefore, for the ex-

perimental analysis of different networks such as citation, authorship, and affiliation network of Italian researchers, the ACM Digital Library[1] was used containing a list of 5250 Italian researchers, 6501 scientific contributions, and 1772 affiliations.

3.2.2 Conceptual Model of Scientific Entities and Communities

In this section is presented a model to represent the data extracted from different repositories into a common meta-data format called Scientific Entity.

Definition of Scientific Entities

A Scientific Entity is an abstract representation of all scientific related content such as journals, papers, conferences, seminars and wikis. In this thesis three types of entities that share common properties is used.

1. *Scientific Contribution:* This concepts refers to any source type of scientific knowledge which includes *traditional contributions*, such as papers, journals and books, and *non-traditional contributions* such as wiki pages, blogs, and datasets. Formally:

$$SC_i = (t, \{P\}, a, L, c, C_t, d, v) \tag{3.1}$$

Where:

- SC_i is the Scientific Contribution i
- t: Title of the contribution.
- $\{P\}$: Set of people related to the contribution(authors, reviewers, editors).
- a: Abstract of the contribution.
- L: Set of labels or keywords of the contribution.

[1] http://portal.acm.org/dl.cfm

- c: Content of the contribution.
- C_t: Set of other Scientific Contribution titles citing the current.
- d: Date of publication.
- v: Venue.

2. *Person:* It represents all types of people involved in the production or dissemination of Scientific Knowledge. Each person has a type depending on the role the perform, as for example reviewer and author.

$$P_i = (n, a, c) \tag{3.2}$$

Where:

- P is the person i.
- n : complete name
- a: affiliation
- c: country

3. *Event:* An Event can be represented as a meeting of people elaborating, listening, discussing, or disseminating scientific knowledge. E.g. Conference or Seminar.

$$E_i = (t_y, n, t_i, d, (P, t_r), SC) \tag{3.3}$$

Where:

- E is the event i.
- t_y : type
- t_i : date of the event
- n : name
- d : description
- (P, t_p) : P represents person, and t_r is the role of the person
- SC : set of Scientific Contributions.

Scientific Community Definition

According to Definition 1 **scientific community** is a set of closely related *scientific entities* that can be identified by a single label or the **name of the community**. In other words, it is a set of characteristics describing the entities of that community.
A scientific community is a labeled aggregation of scientific entities according to a membership function.

$$C_i = (L, (e^{[w]}, t))$$

Where:

- C_i is the Community
- L is the label that identifies the community
- e is a scientific entity that can be any of the following: scientific contribution, person, event or collection
- w is a relatedness coefficient that represents the degree in which an entity is part of the community
- t is a time relation between the entity and the community that could provide the period of time in which an entity is part of the community.

3.3 Conference Network (CN)

Different scientific networks can be built depending of the type of relations between scientific entities. For example, a few but well known scientific networks are:

- **Citation** ($SC <=> SC$): citation occurs when a scientific contribution (SC) has as

reference other scientific contribution to support part of its content. With this network we capture relations between scientific contributions, if one contribution cites another; it could mean that both have similar content.

- **Authorship** ($P <=> SC$): it produces when one or more authors have participated on the elaboration of a scientific contribution. With this network we capture social networks, researcher who published a contribution with another researches is likely to be part of a certain scientific community. The connection between authors in this network is known as co-authorship relation.

- **Affiliation** ($P <=> O$): the affiliation occurs when an author is associated to one or more organizations, such as universities, research centers, companies, and so on.

The detection of communities on these network will provide different community structures and meanings. For example, the citation network will tend to provide topic related communities, while the authorship network will let us get more closely relations inside the community since co-authors often know each other. In 6.1 is described the analysis made using these networks.

One of the main problems is that the vast majority of the clustering algorithms used to detect communities do graph partition on the network [44]. This means that a node only belongs to a particular community after the clustering process. This is a problem if we seek for overlapped communities by their members, especially by authors.

The entity **Person** is part of the affiliate and authorship networks. Hence, after applying community detection clustering algorithms on both network we end up with authors belonging to only one particular community. However, with citation network, we can query authors after the process, but one of the problems is that the correct classification of a scientific contribution depends on the number of citations it has. Hence, it makes the classification/analysis of new or non-cited contributions a difficult task.

In Authorship and Affiliation network we end up with disjoint communities, and the authors belong to only one particular community, while using citation we can query authors after

the process, but one of the problems is that citations appear after a long time. Therefore, is difficult to analyze the current community structure.

In this thesis we propose a new type of network *Conference Network* that will allow us to adjust from disjoint to overlapping communities by query others scientific entities, such as authors, reviewers, scientific publications, among others, in each community.

Definition 2. *A Conference Network is defined as a weighed graph where nodes represent conferences, and the edge between any two different nodes, A and B, is defined by the number of authors that have published in both conferences (A and B).*

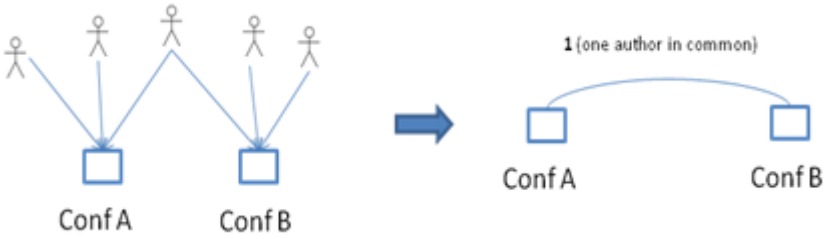

Figure 3.2: Graphical representation of a Conference Network

We create this network because we aim at finding communities of authors who published in the same or similar conferences. In Figure 3.2 is shown a simple example of how an author makes the connection between two conferences.

3.4 Community Detection Cluster Algorithm

In [14][47] a community structure on a graph is defined as a group of vertices which connections within the group are dense, and connections across groups are less dense. See Figure 3.3.

We use the same definition of community structure, and the algorithm for the detection proposed by Girvan and Newman[14] with a weighted graph were nodes represent conferences and the relation between them are defined by the number of authors that have published in

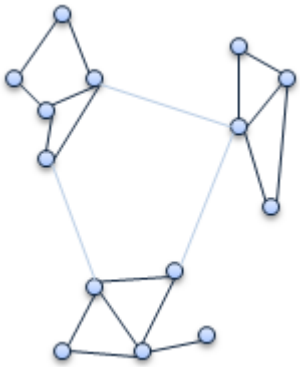

Figure 3.3: Three communities which are densely connected within their vertices, and with a much lower density connection between groups

both conferences. The cluster algorithm is based on **betweenness centralization** which has been studied in the past as a measure of centrality and influence of nodes in networks, first by Freeman[3] who has defined the edge betweenness centrality of a vertex v as the number of shorter path between other pairs of vertices which pass through v, more over is the influence of a node over the flow of information in the network. In order to find edges which are most between other pairs of vertices, Newman generalizes betweenness centralization to edges, and defines the Edge Betweenness of an edge as the number of shorter path of two different nodes that pass through the edge.

The algorithm performs the following steps:

1. Calculate the betweenness for all edges in the network.
2. Remove the edge with the highest betweenness.
3. Recalculate betweenness value for all edges.

The step 2 is repeated till a desired degree of granularity or no edges remain. The betweenness score for all m edges in the graph of n vertices can be calculated in $O(mn)$ time using the fast algorithm of Newman [4]. Therefore, this calculation has to be repeated per each removal of edges, the entire algorithm runs in worse-case time $O(m^2n)$. The algorithm works fine with

unweighted networks, but it does not provide a generic solution for weighted networks. In [32] has been analyzed the algorithm for weighted graph, and a solution of weighted network is proposed which is based on mapping weighted networks to multigraphs. The problem is because the edge betweenness of an edge in the algorithm is defined as the number of shortest paths between vertex pairs. If we define the weight of the edge as a measure of closeness of two nodes, this could mean how related two people are. Then, if we define the length of an edge to change inversely according to its weight , in other words if a node v has a connection of 2 with another node s, it means that will be half as far from those nodes who have connection of 1. Therefore, high edge-betweenness score will fall in strong connections and we will tend to remove edges between well connected pairs, and we want the algorithm do the opposite. Another option is to consider the weight of the edge as it is, in our case represent the closeness of two conferences. Hence, if we calculate the shortest path for a weighted network, the paths will tend to follow weak connections in order to get the shorted path while leaving strongly connected conference in the same community. The solution of the problem summarize as follow: we calculate the betweenness of all edges in our weighted graph considering the weights for the shorted path algorithm. Then, we divide each such betweenness by the weight of the corresponding edge, removing the edge with the highest resulting score, recalculate the betweenness, and repeat. However, we cannot apply this algorithm to all types of weighted graph regardless of the meaning of the edge value. Weight values should represent closeness of nodes, a bigger value of an edge should represent a closer relation of a pair of nodes.

3.5 Measuring the Quality of the Community

The algorithm described in Section 3.4 falls into the category of divisive cluster algorithms, the output produces a dendrogram which represents an entire hierarchy of possible community division of the graph (See Figure 3.4). The possibility of measuring the quality of the structure will help us to select the appropriate cut in the dendrogram, in order words it will let us to select a good partition. For this purpose, in [34] is defined a measure of the quality of a particular division of a network called Modularity, and is defined as follows. Let e_{ij} be the

fraction of edges in the network that connect vertices in group i to those in group j, and let $a_i = \sum_j (e_{ij})$. Then

$$Q = \sum_i e_{ii} - a_i^2 \tag{3.4}$$

Q is the fraction of edges that fall within communities, minus the expected value of the same quantity if edges fall at random without regard for the community structure. If the number of within-community edges is no better that random, we will get $Q = 0$, and values close to 1 indicate networks with strong community structure. However, it is expected in practice to have values in the range from about 0.3 to 0.7. The modularity Q of graph opens a variety of algorithms to detect community structure based on the optimization of Q. However, exact modularity optimization is a problem that is computationally hard [9]. Hence, efficient algorithms must deal with some heuristic in order to get result in polynomial-time. To mention some of them, in [6] is proposed a method which deals with large weighted networks in short time and unfolds a complete hierarchical community structure for the network. Newman also propose a fast algorithm with running time of $O((m+n)n)$ which runs in reasonable times for networks of up to a million of vertices [8].

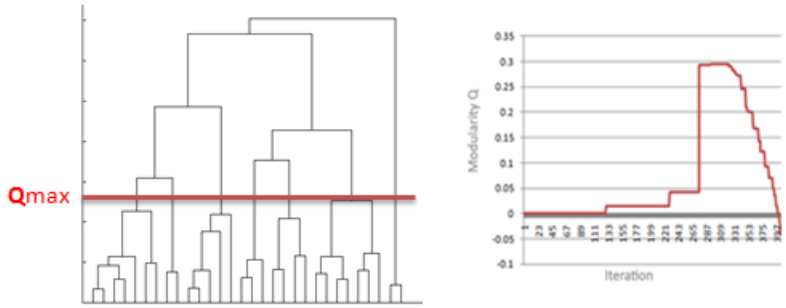

Figure 3.4: Selection in the dendrogram with the highest Modularity Q

The measure of the modularity Q of a weighted graph can also be calculated [32], and is defined as follow: Let c_i be the community to which vertex i is assigned. Then the fraction of the edges in the graph that fall within communities, is

$$Q = \frac{1}{2m} \sum_{ij} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j), \tag{3.5}$$

where c_i is the community which vertex i is assigned, $\delta - function$ $\delta(u, v)$ is 1 if $u = v$ and 0 otherwise, A_{ij} represent the weight of connection from i to j, and $m = \frac{1}{2}\sum_{ij}[A_{ij}]$ is the number of edges in the graph. If we preserve the degrees of vertices in our network but otherwise connect vertices together at random, then the probability of an edge existing between vertices i and j is $\frac{(k_i k_j)}{2m}$, where k_i is the degree of vertex i.

3.6 Building the Community Network

Once communities are detected, we proceed to create a network on top called **Community Network**, this network will allow us to visualize the connection between communities, and to apply some metrics in order to analyze them.

3.6.1 Community Network Definition

$$CN = ((C_i, C_j, O_{ij}))\forall ij, i \neq j \tag{3.6}$$

Where:

- C_i : community i
- O_{ij} : the overlapping from Community i to j

3.6.2 Overlapping

The overlapping/connection between communities is defined by *the percentage of elements two communities share.* If two communities share entities, an edge between communities is created, and the weight is proportional to the number of entities the community has (See Figure 3.6.2).

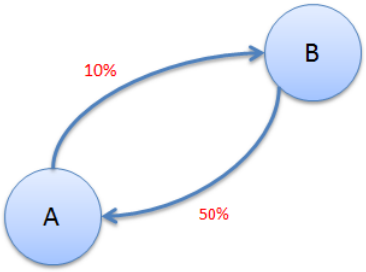

Figure 3.5: An example of the overlapping(edge) between two communities(nodes). Community A shares 10% of their members and Community B shares 50% of their members.

For example, if community A has 100 members, and share 10 members with another community B of only 20 members. Then, the overlapping $O_{AB} = 10$ % and $O_{BA} = 50$ %. The equation 3.7 formalize the definition.

$$O_{ij} = \left| \frac{C_i \cap C_j}{C_i} \right| \times 100 \tag{3.7}$$

The overlapping is asymmetric because communities may have different sizes. Hence, the overlapping value is calculated in both directions.

3.7 Naming Communities

Communities should be identified by a certain name, and it has to characterize the community. This part of the process opens a wide range of possible approaches, which could be used in order to label communities. One possible research line in this field is to use text mining techniques on the literature of the contributions for getting relevant keywords. Some useful tools for the proposed are Leximancer[23], Automap[4], GATE[13], YALE/RapidMinner[37]. The application of these techniques could be done based on the analysis of the abstract, tittle and keywords, which are public available in dblp dump. However, in this thesis we open the field and let to future works a more deeply study that could focus on improve the way scientific communities are labeled.

In this work is used two different approaches. The first method proposed for creating the name of the community is done by using the conferences which are part of the community. The algorithm select the biggest conferences in the community and use the acronym name to label the community. The reason of using the name of the community and not for example the keywords of papers, or other info, is because people(researchers) can infer the topic when looking at the conference, they can identified quickly their community of interest, while providing a topic base name could raise discussion about the correct classification.

Definition 3. *The name of the community is defined by the biggest top-k names of conferences which are part of the community.*

The names of the conferences help researchers to read the topic of the community if they know the conference. Hence, the other approach consist of adding extra information about the topics of the community. We call to this extra information **tags**. Tags could be defined by the user, or imported from external sources.

The tags of communities are keywords, such as information retrieval, database, www, etc., which help to identified the topic of the community. In this approach is used the classification of conferences public available from DBLP[2]. The source contain a list of topics/subjects for a subset of conferences. The algorithm for tagging communities check the classification of conference from DBLP and tag the communities by matching the conferences found in the community with respect to the conference found in DBLP classification.

[2]http://www.informatik.uni-trier.de/ ley/db/subjects.html

Chapter 4

Community Metrics

In Section 2.4 actual methods that measure the scientific productivity and impact of researchers and contributions were analyzed. In this Chapter new community based metrics are presented in order to improve current evaluation methods by using the power of communities.

The lack of context in actual metrics makes unfair comparison of researchers working on different disciplines. The detection of scientific communities will help us to provide a context for normalizing current metrics, and provide the basis to propose new community-based metrics.

In this thesis the **h-index** is used in order to normalize the scientific output of researchers to the community they belong to. However, the metrics proposed here can be easily extended to any other metric.

4.1 Community Metrics

4.1.1 Community Impact C_{IMP}

This index aims at assessing the scientific productivity or possible impact of a Scientific Community, by analyzing the h-index of the community members. One approach for measure the **impact of the community** could be to easily compute the average h-index of the members. The problem of using the average is that the size of a community can differ greatly with respect to another community, smaller communities with a few members and good h-index will tend to have good average, while bigger communities with many authors with high h-index could be affected by the authors which have low h-index (new researchers). In other words, new authors that comes to a community will decrease the community impact, making large communities with many good researchers have a low impact due to the authors who have low scientific output.

Another approach could be to select the most representative members of the community and compute the average (top-k members). The problem here is to select the appropriate k, and also the value is not fixed to the size, which means that the average will not take into account all the members of the community, neither the size of the community.

The metric proposed in this thesis for measuring the scientific impact of the community is defined as follow:

Definition 4. *A community has a scientific impact* $\boldsymbol{n}$ *(*$C_{IMP} = n$*) if* $\boldsymbol{n}$ *of their authors have at least* $\boldsymbol{n}$ *h-index, and the other authors have at most* $\boldsymbol{n}$ *h-index each.*

The **Definition 4** is an extension of the h-index definition to a community context. The C_{IMP} metric considers all the members of the community and is fixed to the size of the community because a C_IMP of n needs at least n authors with at at least n h-index. Figure 4.1 shows how the C_IMP value is calculated. For example, let suppose we have a group of 15 authors in **Community A**, and a group of 100 authors in **Community B**. Both community are related to the same topic and we are interesting to find a community with high impact for searching scientific content. **Community B** has 30 researchers with high h-index (at least 35 h-index each), another 30 members with middle impact (between 10-25) and the rest 40

members with an average of 5 h-index. **Community A** is a very small community and all the members have an average of 35 h-index. Hence, if we compare the groups, community B has more researchers with high h-index than community A, and therefore community B is the one we are interested to search on. But if we calculate the average impact, the community A will get a higher value, and similar values is obtained if we use top-10 average. While the metric proposed here will give a value of 35 to Community B, while community A will get a value of 15.

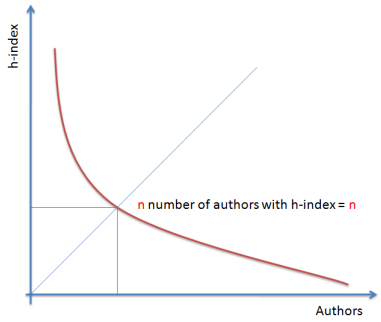

Figure 4.1: Community Impact Metric

4.1.2 Community Health C_{HT}

Communities which are not well connected with others communities (known as **closed communities**) do not help to the transference of knowledge, nor the dynamic of the community. In the opposite, a community that shares members in many other communities will tend to have a good transference of knowledge, and will help to the dynamic of the members (new members coming). this type of community is defined as a **healthy community**.

The metric proposed here measure the healthy of the community, and is defined as follow:

Definition 5. *The Health of a community C_{HT} is defined as the number of communities that share authors in common (overlapping).*

Let C_N be the community network, where nodes represent communities, and edge represent the overlapping defined by the number of shared authors. Hence, the Healthy of a Com-

munity A is equal to the $degree(A)$. The degree of a node is defined as the number of incomming/outcomming edges that the node has.

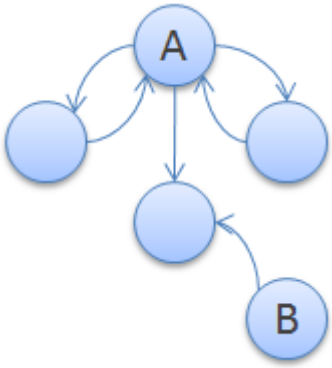

Figure 4.2: Community **A** with a high degree of overlapping makes the community healthy, while Community **B** with a low degree makes the community unhealthy or closed.

4.2 Author Metrics

4.2.1 Author Membership Degree A_{MD}

Is important, when talking about the members of the community, to analyze the membership degree of authors. If an author has published in the Community A 10 papers, and only 1 paper in the Community B, it is unfair to consider the same degree of membership, especially when analyzing the impact of the community.

The following metric captures the membership degree of an author to a community with respect to his publications.

Definition 6. *Let $|C_{A_i}|$ be the number of contributions of author A in the community i, and $|C_A|$ the total amount of contributions of author A. Hence, the authorship degree of author A in community i is defined as: $A_{MD}(A_i) = \frac{|C_{A_i}|}{|C_A|}$*

The value is the total number of publications an author has in the community with respect

to his total number publications. With this metric, a threshold can be defined for computing metrics. For example we can consider for computing the C_IMP only authors with a membership degree greater than 0.3.

4.2.2 Author Community Context h-index A_{CH}

This metric measures the impact of an author by normalizing it to the community. Let suppose that a researcher A has published 5 papers in **Artificial Intelligence (AI)**, and other 20 papers in **Data Mining Community (DM)**. All of them have contributed to have an h-index of 20. In the other hand, let suppose that we have another researcher B with 15 publications in AI and a few others publications in other communities, having a h-index of 15 by the 15 publications on AI. Now, if we compare without put in context the community, the scientific output/impact of author A is greater than author B. But, is not the case in AI community where author B has a better impact.

The following metric provides context to the h-index to the communities researcher belongs to. This index will mainly help us to select appropriate researchers on certain fields when looking for recruitment, rank, or compare researchers for different purposes.

Definition 7. *Let C_{A_i} be the set of contributions of author A in community i. The A_{CH} of author A is defined as the $\boldsymbol{n}$ number of contributions in C_{A_i} that have at least $\boldsymbol{n}$ citations, and other contributions in C_{A_i} that have at most $\boldsymbol{n}$ citations each.*

4.2.3 Normalized h-index $\overline{A_{CH}}$

The h-index of two researchers working on different communities/disciplines is normalized by $\overline{A_{CH}}$. Have a h-index of 10 in Biology does not has the same meaning than a h-index of 10 in Computer Science, and also in many sub-fields. With the Community Impact C_{IMP} metric we capture the scientific productivity of the community, and therefore the h-index of a researcher can be normalized to the community he belongs to.

Definition 8. *The Normalized h-index of an author* $\overline{A_{CH}}$ *is defined as the fraction of the Author Community Context h-index* A_{CH}*, and Community Impact* C_{IMP}*.*

The definition describes the following equation: $\overline{A_{CH}} = \frac{A_{CH}}{C_{IMP}}$

All the metrics proposed in this thesis was implemented by the Community Engine Tool, and this approach is part of the largest effort aimed at improving the way scientific content and researchers are evaluated.

Chapter 5

Community Engine Tool (CET)

The Community Engine Tool (CET) is a desktop application that was designed and developed in order to support the requirements for all the process, previously described, of the detection and evaluation of scientific communities. This tool is part of the **Community Discovery Module**, which is one of the large set of components of the LiquidPub architecture.

5.1 LiquidPub Architecture

The work in this thesis is part of a big puzzle of component that form the LiquidPub Project. In this section we will describe in a nutshell the LP architecture in order to better understand how the Community Module fits into the LP platform.

Figure 5.1 gives an overview of the multi layer architecture of the platform, which allow to build the components with different degrees of abstraction. In the lower layer is the *Basic Services* layer, in the middle the *LiquidPub Services*, and on top the *Scientific Dashboard or Web Interface*.

The **Basic Services** layer provides uniform access to data and meta-data from different sources, such as blogs, wikis, springerLink, citeulike and so on. It receives requests from the

Upper Layer mainly through internal low-level API, and query the requested source/s. The specifics and the code at this layer are obscured from the outside, so they are only accessible through the low-level API calls.

The Middle-Level Layer (**LP Services**) contains the components that allow the creation, evaluation and dissemination of scientific content. The interaction between these components are done throw REST/SOAP, and it provides the necessary services for the scientific dashboard. The **Community Discovery Module (DCM)** fits in this layer and export the services to improve search and assessment of scientific content and researchers.

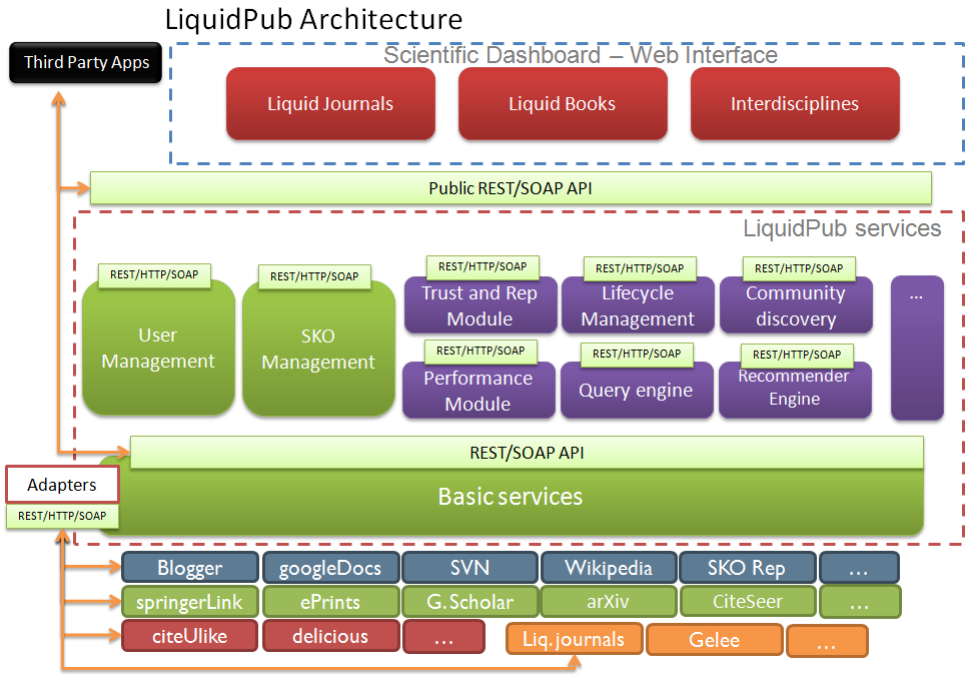

Figure 5.1: Architectural overview of the LiquidPub core platform

A list of tools that DCM interacts with are:

- Liquid Journals: one of the services offered by the community discovery module is to provide a diversified search of scientific content by using the power of communities. This service is used by Liquid Journal tool in order to allow an alternative way to search scientific contributions.

- Reseval: the DCM interface with this component in order to get metrics of authors and contributions.
- Group Comparison Tool: it loads the groups into the CET tool to visualize and analyze them in the Community Network context. Also, the discovered communities can be exported to this module in order to apply group comparison metrics through the web.

5.2 Community Engine Tool Architecture

The architecture of the Community Engine Tool is composed of five main components:

1. **Network Manger (NM)**: this module manages the transformation of the source data into a network of conferences. All the pre-processing steps are done in this module.

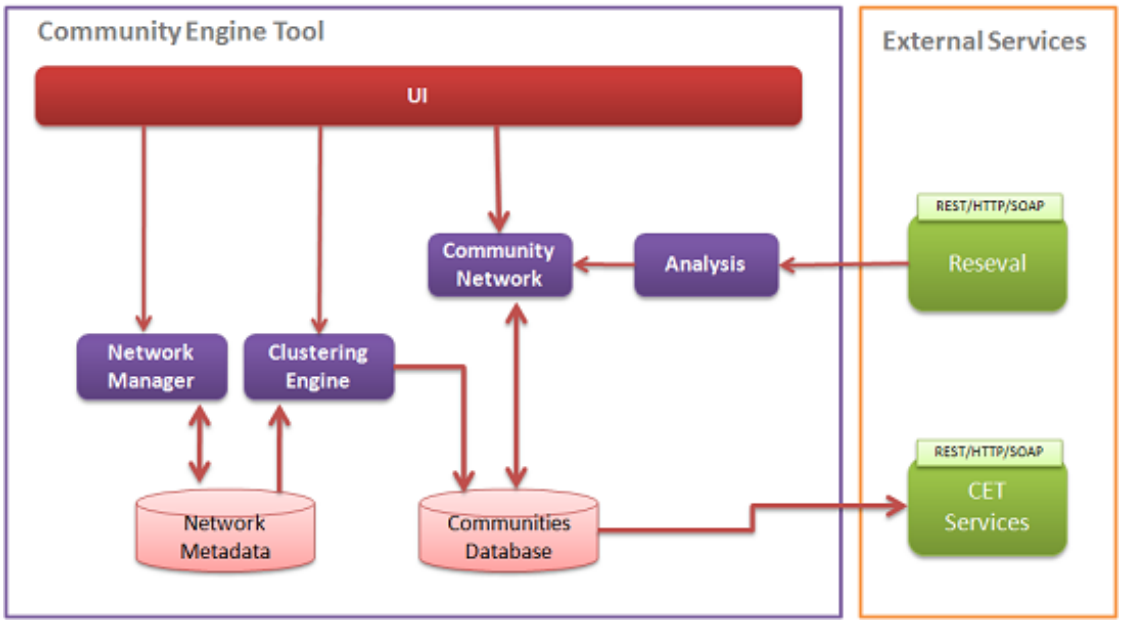

Figure 5.2: Community Engine Tool Architecture

2. **Clustering Engine (CE)**: all the community detection cluster algorithms are built in this component. The network of conference is received as input, and user defined cluster algorithms are applied in order to finally obtain cluster of conferences.

3. **Community Network (CN)**: this component manages the complete creation of the CN, the members of each community, and the overlapping between them based on the obtained cluster of conferences.

4. **Analysis**: this module analyzes the CN, it interfaces with the Reseval tool by calling its REST services in order to get author metrics such as h-index, g-index, and total citation count. The communities and people are analyzed in this component.

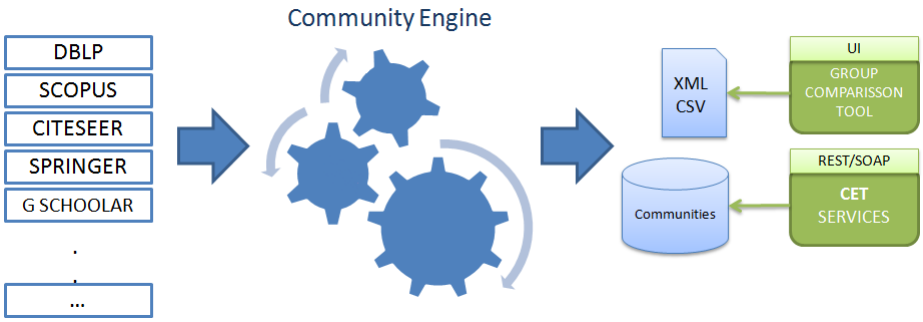

Figure 5.3: The figure shows the complete process of the Community Module

In Figure 5.3 the complete process of the Community Module and the role of the Community Engine Tool are shown. The tool processes the source data and exports the discovered communities into a database or csv/xml file in order make them available for other components of the LiquidPub Platform.

5.3 Services

The current version of the Community Engine Tool prototype supports the following services:

1. **I/O Manager:** this service is responsible for loading and export data in different formats, supported by other components of LiquidPub project. More details in Appendix A.2.

2. **Detection of Scientific Communities:** this service applies clustering algorithms

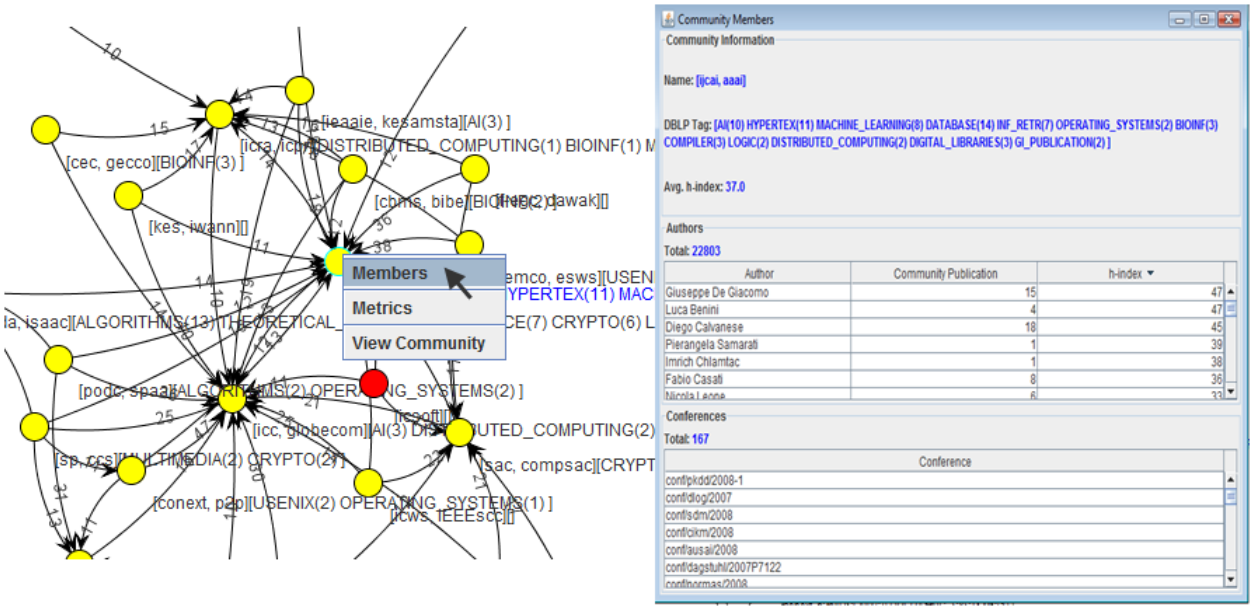

Figure 5.4: Member details of the selected community

for the detection of scientific communities. The tool implements the algorithm proposed in Section 3.4.

3. **Creation of the Community Network:** it provides the graph representation of the discovered communities, where nodes represent communities, and edges correspond the overlapping between communities. Figure 5.6 shows the community network created by the tool.

4. **Naming process:** this service manage the creation of the community name by analyzing the conferences which are part of the community, and the tag aggregation to communities using DBLP conference classification. See Figure 5.4 for more details.

5. **Analysis:** this service plot the h-index distribution graphs (Figure 5.5) showing metrics for authors (h-index, g-index, publication number, citations, and so on), and the metrics defined in Chapter 4 such as COM_{IMP}, C_{HT}, and A_{MD}.

6. **Visualization:** this service deal with the visualization of the Community Network, and the network inside the Community.

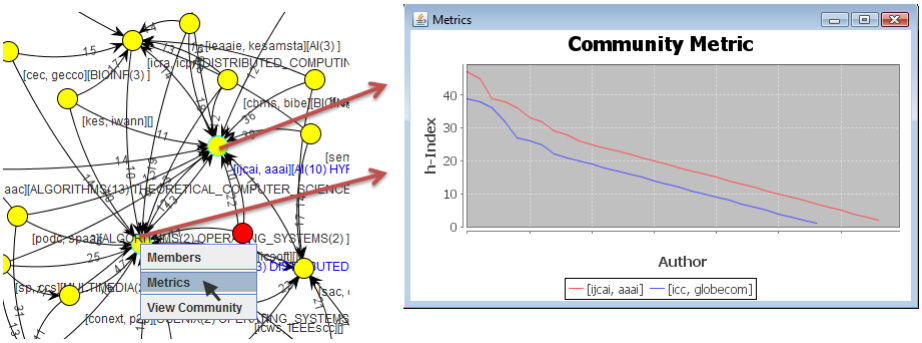

Figure 5.5: The graph showing h-index distribution of the two communities, selected from the community network in the CET tool

5.4 Related Tools and Implementation Details

Before developing the tool, we reviewed similar tools already available. In the following, the list of the analyzed tools (with brief explanation for each tool) is presented

- ORA: it is a dynamic meta-network assessment and analysis tool developed by CASOS at Carnegie Mellon [11]. It contains many SNA metrics that will help the detection of scientific communities.
- Weka: is a tool made for data mining task and the algorithms are focused on machine learning techniques [45].
- Igraph: is free software packages for creating and manipulating graphs. It includes implementation of graph theory problems and network analysis methods [17].
- JUNG: is a software package that provides suppport for the modeling, analysis, and visualization of data that can be represented as a graph or network. It is written in Java, which allows JUNG-based applications to make use of the extensive built-in capabilities of the Java API, as well as those of other existing third-party Java libraries [19].

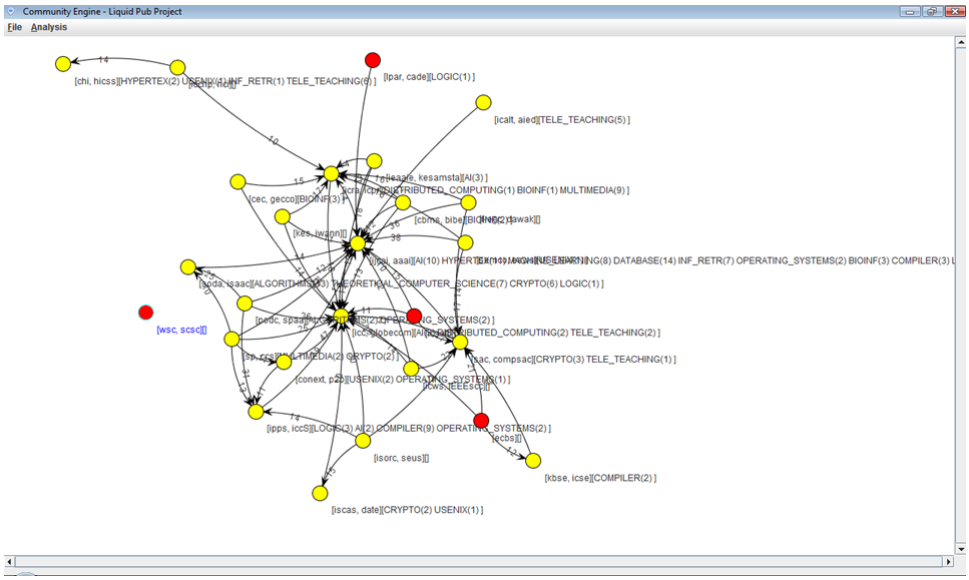

Figure 5.6: The Community Engine Tool (CET). Community Network obtained after detecting communities on DBLP dataset 2007-2009

ORA and Weeka are ready tools that can be used for our purpose. In fact, the analysis of different networks such as citation, affiliation and authorship were made using ORA (more details in Chapter 6). Weka is an open source project written in Java and it has implemented a set of clusters algorithms, such as the well known k-means, that we can use for the sake of detecting communities. However, the tools is oriented on data mining process, and it does not provide SNA techniques which we are interesting on.

ORA tool is a good starting point to analyze networks, it provides many SNA techniques that helps the analysis of different types of networks. However, it is a closed-source project under a Freeware for non-commercial use license. Hence we can not adapt the tool to our requirements, neither integrate with other LiquidPub services.

The need of a tool that can be easily integrate with the LP platform and let us modify according to our needs has taken us to design and develop the CET tool. One of the first step of the implementation was to decide which framework use as core for graph management and visualization. In this field igraph and JUNG suite our requirements. However, we select JUNG because is written in Java as well as all the components of the LP Platform, and thus

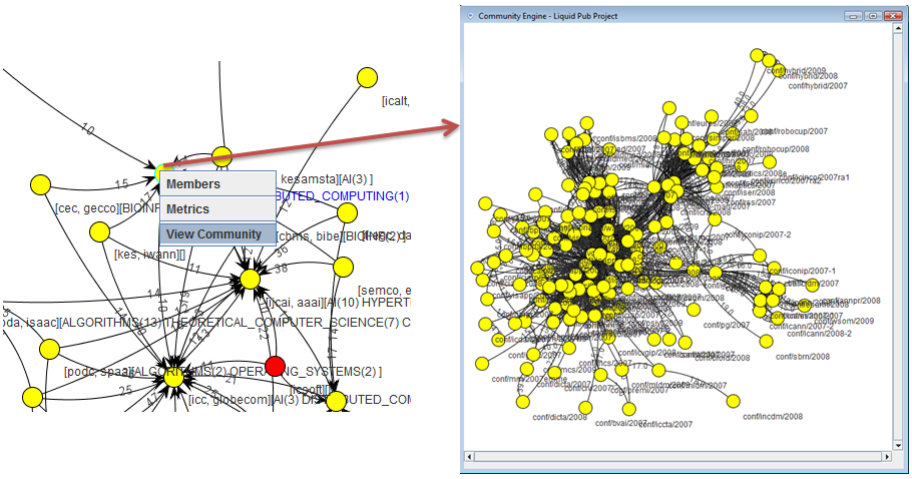

Figure 5.7: The Conference Network of a selected Community

it facilitate the integration and maintenance by the members of the group, while igraph can be used only as a library in C/C++, R and python.

The tool is completely written in Java and use JUNG framework [1] for graph management, and Maven[2] for project management and build automation. In Appendix A.1 the organization and details of the source files (packages) are detailed.

The Community Engine Tool still in development phase, and more functionalities are intended to add in the next beta version such as the evolution analysis of communities and authors. It also desired to make the source code public available once a release version is reached.

[1] http://jung.sourceforge.net/
[2] http://maven.apache.org/

Chapter 6

Results and Validations

This Chapter describes the experiments made on different scientific networks, such as citation network, authorship network, and affiliation network. Then, the application and validation of our algorithm and metrics using the Community Engine Tool.

The main objectives of the experiments are:

- Analyzing scientific networks, using existing techniques and tools to detect communities, for the final goal of proposing new techniques and algorithms for the detection of communities.
- Validating the proposed techniques and algorithms.

6.1 Analysis of Different Scientific Networks

One of the first step in this research is to analyze different scientific networks in order to better understand the structure of them, and to test current cluster methods for community detection.

Each network has different structure and meaning, for example building communities based on

authorship could represent groups of people that work together, while building communities based on citation network could detect topic based communities. Also clustering different network structures will help the research to provide new techniques and algorithms for the detection of scientific communities.

This section describes the analysis on different scientific networks, which are based on the type of relation between Scientific Entities (SE). The analysis consist of taking each network separately and apply cluster algorithms in order to build communities based on each network. Then we will combine each network into one complete-network and perform the same analysis.

6.1.1 The Input and Pre-Processing

For this experiment, a list of 5250 italian researchers was used to match with Digital Library ACM [**?**]. This dataset is selected in order to analyze the result obtained with the people of University of Trento (Italy). The dataset contains: 1289 people found, 6501 contributions, 1772 affiliations, and more than 25000 relations between them.

The data obtained is saved into a ER database schema. Each type of Network is consulted from the database and transformed into a DyNetML XML format, which is compatible with ORA (see Appendix B.1). The results of this process are three XML files representing the three types of networks (citation, affiliation, and authorship) which are used as the input of ORA, which is a dynamic meta-network assessment and analysis tool developed by CASOS at Carnegie Mellon [11].

6.1.2 Citation Network

Citation occurs when a scientific contribution has as reference other scientific contribution to support part of its content. This network captures the relation of scientific contributions, and therefore it provides topic related (similar content) clusters of contributions. Figure 6.2 shows the citation graph obtained in this experiment.

Node-Level Measure	Avg	Stddev	Min/Max	Min/Max Node
Centrality, Hub	0.1292	0.2946	0.0000	726 nodes (66%) have this value
			1.0000	76 nodes (6%) have this value
Centrality, In Degree	0.0020	0.0025	0.0000	371 nodes (33%) have this value
			0.0238	Hardware-software co-design of embedded systems
Clique Count	0.4670	1.2381	0.0000	832 nodes (76%) have this value
			13.0000	Tools and approaches for developing data-intensive Web applications

Figure 6.1: Node-Level Measure

Analysis

Figure 6.1 details the analysis made to the network. The contribution *SC Tool and approaches for developing data-intensive Web applications* has the highest score on *Clique Count* measure. We take this node and analyze the *Sphere of Influence* in the Citation Network with a radio of 5. Figure 6.3 shows the graphical result. In the graphic we can see how citation gives as the similarity of content and how two topics are connected within the citation graph. For example, in the figure we can see that a center node *Conceptual Database Design* divide two topics: Web and Data Base.

Clustering

For cluster process, the Newman's Clustering Algorithm, CONCOR Structural Equivalence Algorithm , Clique Detection Algorithm, and Johnson Hierarchal Clustering Algorithm are used. All of them have been implemented by ORA. In the following the statistical results for each algorithm is presented.

Newman

- **Groups Founds**: 187
- **Min Size**: 1

Figure 6.2: Citation Graph. A Scientific Contribution is represented as yellow node (knowledge) and edge represent the citation between two contributions.

- **Max Size**: 62
- **Average**: 5.839
- **Stddev**: 9.9161

CONCOR Structural Equivalence Algorithm

- **Level**: 1
- **Groups Found**: 2
- **Min. Size**: 221
- **Max**: 871
- **Avg**: 546
- **Stddev**: 459.61

Figure 6.3: Sphere of Influence of Scientific Contribution Tool and approaches for developing data-intensive (red node) with radio 5.

Clique Detection Algorithm

- **Minimum clique size**: 3
- **Cliques Found**: 159 cliques.
- **Min. Size**: 3
- **Max**: 5
- **Avg**: 3.20
- **Stddev**: 0.4221

Johnson Hierarchical Clustering Algorithm

- **Number of groups**: 30 (input)

- **Min Size**: 1
- **Max Size**: 1043
- **Avg**: 36.4
- **Stddev**: 190.12

In this analysis, after comparing different clusters results, especially in the standard deviation and the number of groups found by each of them, we conclude that Newman algorithm makes the best distribution of the contributions into communities, and the majority of the groups have similar topics. One point of discussion with this network is the meaning that the community represents when it is build based on citation network. People that belong to a community have not social interaction with other members of the community, such as events or a co-authorship that could capture this relation.

Another week point of using this network is that new contributions do not have citations, and they start coming (if they come) after new publications arrive, making the detection of communities for new and un-cited contribution a difficult task.

6.1.3 Authorship Network

Authorship Network denotes when one or more people have participated on the elaboration of a scientific contribution. With this network we capture social networks, researcher who published a contribution with another researchers is likely to be part of a certain scientific community. A filtered visualization (hiding isolates nodes) of Authorship Network is shown in Figure 6.4. Red nodes represent authors and yellow nodes represent contributions.

As an example, the *Sphere of Influence*, which represents the connections of the node Luca Benini is shown in Figure 6.5. This graph shows also a co-authorship network as well. For example in the graph we can see that Luca Benini co-authored papers with: Bertozzi, Bogliolo, Acquaviva, Dalpasso, and Favalli.

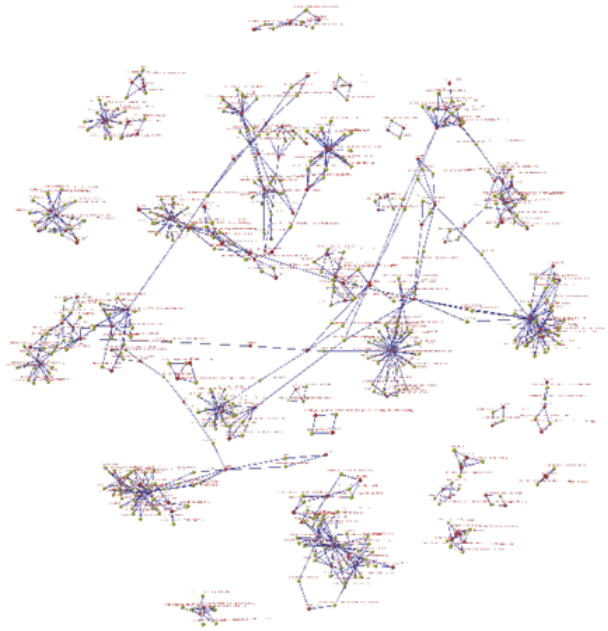

Figure 6.4: Authorship Network

Clustering

Newman

- **Groups Founds**: 309
- **Min Size**: 2
- **Max Size**: 157
- **Average**: 11.73
- **Stddev**: 21.3

CONCOR

- **Groups Founds**: 2

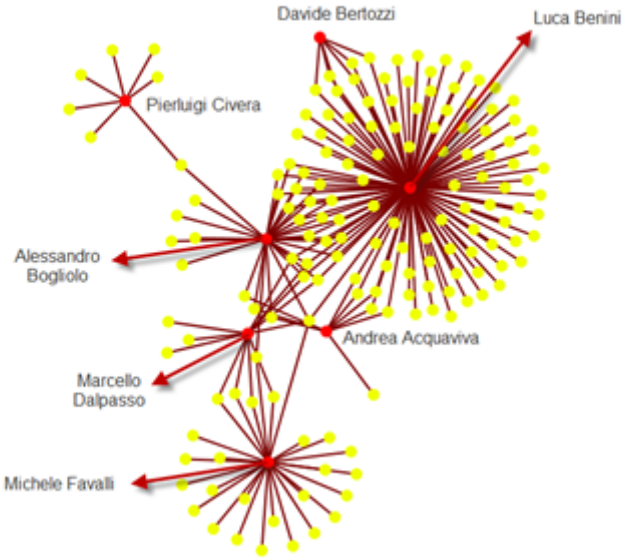

Figure 6.5: Sphere of Influence of Luca Benini with radio of 5 in Authorship Network. Yellow nodes are SC and red nodes are People. Edges between them represent an authorship.

- **Min Size**: 504
- **Max Size**: 3123
- **Average**: 1813.5
- **Stddev**: 1851

Newman algorithm has better result in this analysis, only 2 groups with a very high standard deviation have found by the CONCOR algorithm. The groups found by Newman in this experiment were analyzed by people from the University of Trento in order to check if the algorithm makes a meaningful distribution. We conclude that Newman performs well the detection based on the data we select. However, as for the network, the network contains people, and therefore at the end of the process they only belong to one particular community, which is a problem if we seek to build overlapped communities by their members.

6.1.4 Affiliation Network

Affiliation occurs when a person is member of an organization, such as Universities, labs, and so on. An author can have more than one affiliation. A filtered visualization (hiding isolates nodes) of the Affiliation Network is showed in 6.6

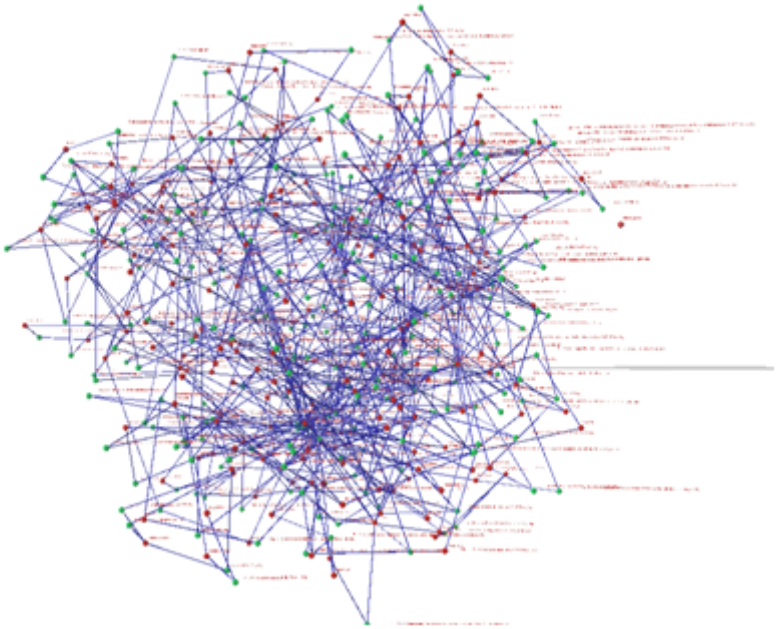

Figure 6.6: Affiliation Network. Red nodes represent people and green nodes represent organizations

We select the node "University of Trento, Trento, Italy" and compute the Sphere of Influence with a certain radio. The result is showed in 6.7

Clustering

Newman Cluster Algorithm

- **Groups Founds**: 341
- **Min Size**: 1

Figure 6.7: Sphere of Influence of the node "University of Trento, Trento, Italy". Green nodes represent organizations and red nodes represent people.

- **Max Size**: 188
- **Average**: 6.921
- **Stddev**: 17.94

In this analysis the Newman algorithm has also better output according to the statistic values. The other algorithms do not provide meaningful result. The result of these experiments has been also analyzed by the people working at the University of Trento (Italy), and they have concluded that the algorithm did make significant group distribution of authors according to their affiliation. However, the network has the same overlapping problem as the authorship network.

6.1.5 Complete Network

In this section all the different scientific networks previously analyzed (citation, authorship and affiliation network) are merged into a single graph called **Complete Network**. The

combination of them is made by connecting each entity node with his current pair in the other network. Figure 6.8 shows graphically how these networks merge into one single network.

After the creation of the Complete Network, Newman's Cluster Algorithm and other analysis are performed in order to find communities.

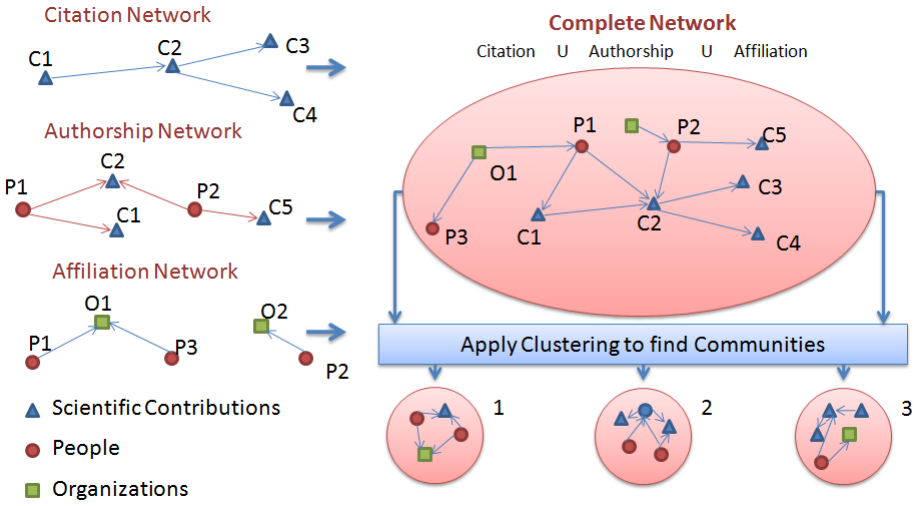

Figure 6.8: Complete Network obtained after merging others scientific networks

We selected some entity nodes, and perform the Sphere of Influence in order to show the network and their structure more closely.

As an example, Figure B.5 presents the Sphere of Influence of node: "Fausto Giunchiglia" with a certain radio of influence.

In the following the statistical description of the cluster process is provided. The result of this experiment has as result balanced communities with respect to each others. However, the meaning of these clusters is not clear because all networks are combined.

Clustering

Newman Cluster Algorithm

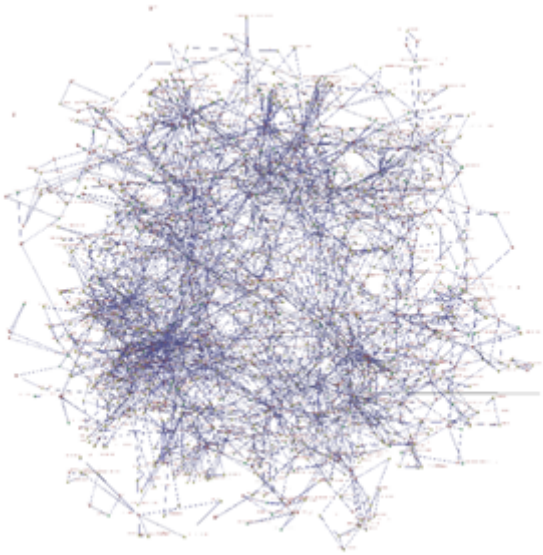

Figure 6.9: Complete-Network of Entities.

- **Groups Founds**: 487
- **Min Size**: 1
- **Max Size**: 434
- **Average**: 12.72
- **Stddev**: 37.95

In conclusion, Newman's algorithm, which is based in edge betweenness index, has proved to be robust in all the analysis, providing significant results in comparison with other studied algorithms. About the networks, all of them provide different community meanings. In this field the affiliation and authorship network capture communities with some social interaction between their members, while citation does not. However, the affiliation and authorship have the problem of classifying a person in only one community when applying graph-partitioning techniques for the detection of communities.

The results of these analysis have been taken to the creation of a new scientific network called Conference Network, that allow us to adjust from disjoint to overlapping communities

by query others scientific entities, such as authors, reviewers, scientific publications, among others, in each community. Also, it has conduced to the study and subsequent implementation of centrality indices in order to identify communities.

6.2 CET Tool - Detecting Communities

This section describes the experiments done with the Community Engine Tool (CET). In order to validate the algorithm, we make two experiments that involve computing our algorithm on dblp data set and compare the community structure obtained with the manual topic classification of conferences done by DBLP.

6.2.1 Topic Classification Analysis

As it was mentioned before, we start by creating a network of conferences which an edge of two conferences is defined by the number of authors that have been published in both conferences. Our goal is to detect communities of authors that have published in the same or similar venue, and compute the overlapping of communities which is defined by the number of common authors. For testing the algorithm, we use the classification of conferences by topic from DBLP [10] in order to group conferences of a particular Topic. Then, we select two groups and compare the resultant community structure with the DBLP classification. The list of conferences and their topic from DBLP are listed in Figure 6.1.

For the **first experiment**, we select conferences of *Information Retrieval* and *Hypertext*, and create the conference network (see Figure 6.2). As we mentioned before, in this network a node represents a conference in a particular year.

The algorithm produces an entire hierarchy of possible community division of the graph, we calculate the Modularity of each partition during the process in order to select the structure which represent the best partition. Good values of Modularity are given on Iteration 294 for HT/IR, and on Iteration 41 for AI/CRYPTO.

Network	Conferences
ARTIFIVIAL INTELLIGENCE	IJCAI, AAAI, EC, AI, UAI, KI/GWAI, IEA/AIE, PRICAI, AUS-AI, EPIA, KR, AGENTS, AIIA, AIMSA, SCAI
INFORMATION RETRIEVAL (IR)	ACM SIGIR Conf, TREC, IEE-ADL, ACM-DL, ECDL, HIM
HYPERTEXT (HT)	HYPERTEXT CONFERENCE (HT), ECHT, UK HYPERTEXT CONFERENCE, GERMAN HYPERTEXT, INFORMATION RETRIEVAL, MULTIMEDIA CONFERENCE, ACM DL, SIGIR, SIGMOD, VLDB, ICDE, DEXA
CRYPTOLOGY/SECURITY (CRYP)	CRYPTO, SUROCRYPT, ASICRYPT, FSE, PKC, INFORMATION HIDING, CCS, RBAC, CHES, SAC, AES, ACISP, CSFW, INDOCRYPT, D-A-CH SECURITY

Table 6.1: The table shows four different topics with a group of conferences that belong to the topic

Network	Conferences	Relations
HYPERTEXT / INFORMATION RETRIEVAL (HT/IR)	33	808
ARTIFICIAL INTELLIGENT / CRYPTOLOGY-SECURITY (AI/CRYP)	80	2208

Table 6.2: Details of the two networks. The number of Conferences corresponds to the number of conferences that have a relation to at least one conference

Let's analyze the community structure found on the highest value of Q. For AI/CRYPTO the algorithm divide the network in two communities and the members of the community match exactly with the classification of DBLP, we have one community with all conferences of the

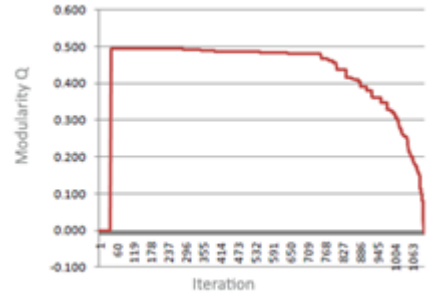

Figure 6.10: AI/CRYPTO cluster process. Best value of Modularity Q is on iteration 41 (0.288511).

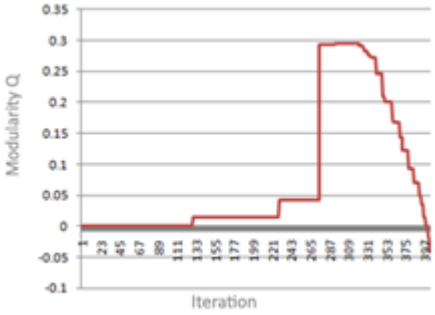

Figure 6.11: HT/IR cluster process. A peak of Modularity is obtained on iteration 294 (0.295608)

Artificial Intelligent (on different years), and another community of with all conferences of Cryptology-Security (different years). The overlapping between the communities is defined by 40 authors. Therefore, those communities are densely connected within their members and not as much connected between them.

As for the community structure found on HT/IR is quite different, this two groups seems to be more related, only one small community of Hypertext is not related to any community of Information Retrieval which is sigmod/2008dbtest. The number of division is equal, we have 4 communities for HT and 4 communities for IR Fig. 8.

Within the two topics that are not very related (CRYPTO and AI), the tool produces the exactly same human classification done by DBLP. In the other hand, the others two topics

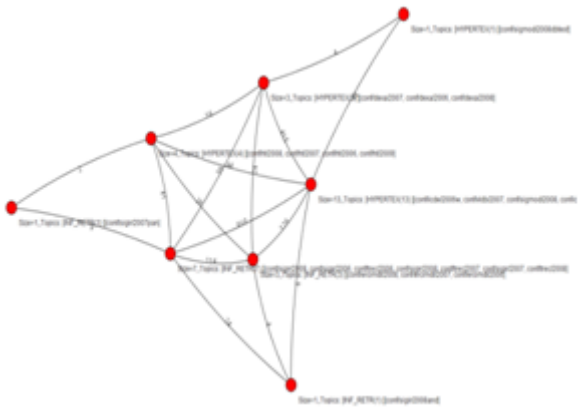

Figure 6.12: HT/IR analysis. Communities found on Iteration= 294 ($Q = max$). The size is defined by the number of conferences, and the tag HYPERTEX and INF_RETR represent the topics in the community based on DBLP classification.

that are more related (HT and IR) the tool outputs an equal distribution of communities with respect to the dblp topic classification. Therefore, it has been demonstrated based on our experiments that the tool produces automatically topic based communities close to human defined classification.

6.2.2 Metric Analysis

In the previous section we validated the community detection algorithm by comparing the result with the DBLP classification. In this section we will apply the metrics proposed in Chapter 4 on the discovered communities and analyze the results.

Data Set

For this experiments, we use the complete list of proceedings, in-proceedings, and authors between 2007 and 2009 from DBLP dump. This dataset will allow us to compare different communities, within the computer science scope, with different impact/productivity.

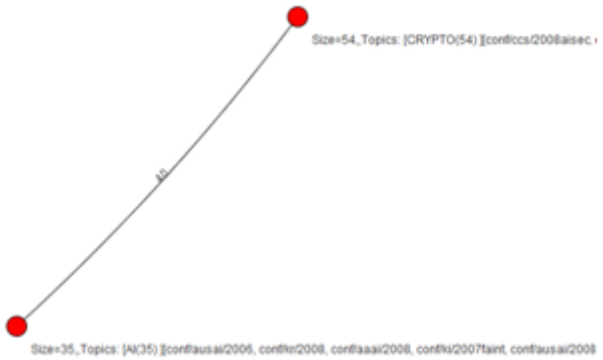

Figure 6.13: AI/CRYPTO analysis. Communities found on Iteration= 41 (Q=max). The size is defined by the number of conferences, and the tag CRYPTO and AI represent the topics in the community based on DBLP classification. The Overlapping between them is 40 authors

Community h-index Distribution

For each community the h-index of all the members is calculated by the tool. Figure 6.14 shows the h-index distribution of each community, its abscissa is ordered by authors with higher h-index first.

The chart shows high values for community *www-csa* and *sac-compsac*, while low distribution for *wsc-scsc* and *conext-p2p*.

In Table 6.3 the healthiest discovered communities are detailed with their values. The lowest values correspond to isolated communities (See Table 6.4).

The h-index of the members is obtained by interfacing with RestEval services. Therefore, the accuracy and complete computation of all the members will depend on the information that is already computed by ResEval. However, we let for future work the incorporation of others services that will help the analysis of community members and their community itself.

Table 6.4 list closed communities found by the tool. These unhealthy communities have

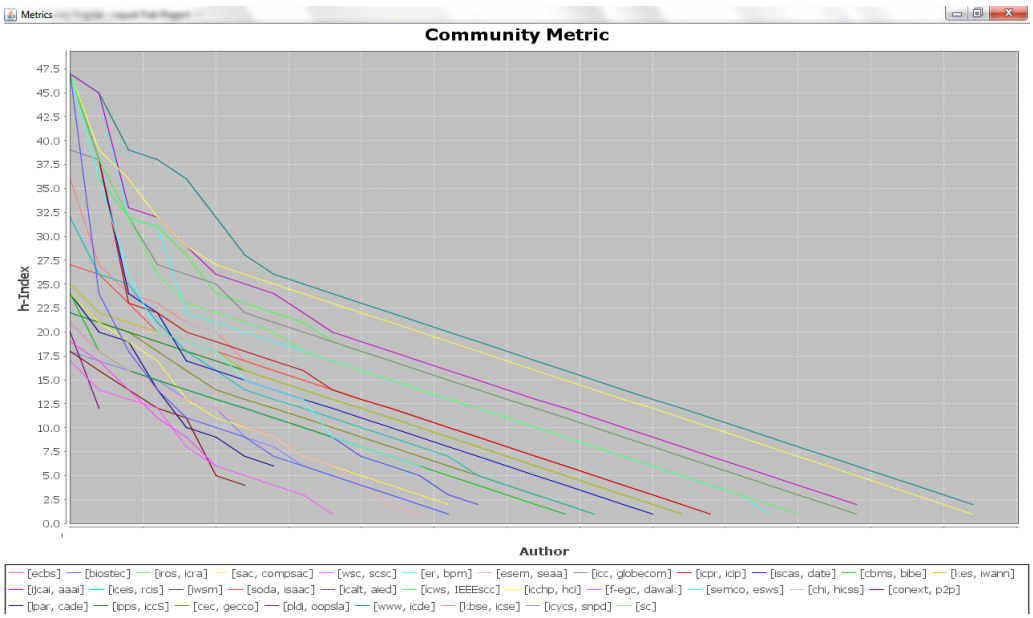

Figure 6.14: Community h-index distribution

Community	C_{HT}	*Authors*	$Avg_{topK}-Hindex$	C_{IMP}
sac-compsac	13	10923	29	20
www-icde	11	15665	33	20
sc	7	74	-	-
icc-globecom	6	26517	29	18
er-bpm	6	662	22	13
icws-IEEEscc	6	2860	27	15

Table 6.3: Healthier communities and their scientific impact

members that only published in their community and not in another. The community *iros-icra* has an important Community Impact value of 17, but it is not as healthy as others communities with similar size and impact such as *sac-compsac*, which is a little bit smaller than *iros-icra*, but it has a healthy value of 13 and a C_{HT} of 20.

Communities *iscas-date* and *iros-icra* are two big closed communities. Conferences **IROS** and **ICRA** correspond to *Robotics and Automation* topic, and the conferences **ISCAS** and **DATE** correspond to *Electronic Circuits and nano-technology*. Hence, the healthiness of

Community	C_{HT}	$Authors$	$Avg_{topK}-Hindex$	C_{IMP}
wsc-scsc	0	2348	8	6
iscas-date	0	14069	22	13
icalt-aied	0	3129	14	11
iros-icra	0	11047	24	17
biostec	0	1931	14	8
kes-iwann	0	4450	18	13

Table 6.4: Isolated communities and their scientific impact

these communities proof that researchers working on these topics are not interdisciplinary, they only published in their community, unlike communties with higher healthy value. However, this topic can open a research line based on the healthiness of the community in order to detect closed topics with a deeper analysis on the topics/fields of each community.

In summary, it is demonstrated that each community has different h-index distribution, which means that the scientific productivity differs in each community, making our metrics a fairer evaluation.

The healthiness of the community helps to identified closed/unhealthy and open/healthy communities. We found important difference in their healthiness between communities with similar scientific impact and size. With this metric, many search algorithms can be proposed based on these values, such as the interdisciplinarity of authors, or diversity of content.

Chapter 7

Conclusions

The work in this thesis contributes significantly to the community and in particular to the LiquidPub project. To conclude this work, the proposed objectives in this thesis and final contributions are summarized, followed by a description of the different research lines that opens the development of this thesis for future work.

The analysis in the current way scientific contents and researchers are assessed motivates the investigation in order to demonstrate that the lack of content produces unfair evaluation when comparing researchers from different communities. Moreover, providing a social network context , new evaluation metrics and search mechanisms based on communities can be proposed.

The main goal of this thesis is to improve **search** and **assessment** of scientific knowledge and authors by providing a **model** and a **tool** that support the detection and evaluation of scientific communities. Moreover, the thesis aims at proposing **new metrics** for the evaluation of individual productivity by normalizing it to the community.

The compliments of the proposed objectives will contribute to reach the proposed goals of the LiquidPub project (described in Section 2.1).

A brief summary of the main contributions, according to its objectives, reached by the thesis

is described as follows:

1. **A model and a complete process for the detection of scientific communities using a conference network**: the proposed model in Section 3.2.2 allows to address the complete list of problems (problems stack) that involves the detection and creation of scientific communities.

2. **An algorithm for the detection of scientific communities**: the analysis of current algorithms used for the detection of scientific communities has helped the correct selection of algorithms and techniques that can be used in order to improve the detection process. The algorithm proposed in this thesis has demonstrated that the discovered communities by the algorithm are close to a human classification.

3. **New metrics for the evaluation of communities to improve the way scientific contents and authors are assessed**: the proposed metrics and the experiments obtained in Section 6.2.2 demonstrate the power of using the community network when analyzing people or content. In addition, the experimental results shows that each community has different scientific productivity, and therefore it makes current metrics unfair comparison of researchers without considering the community they belong to.

4. **An application that supports the detection and evaluation of scientific communities**: this thesis has yielded the development of the **Community Engine Tool** (CET) detailed in Chapter 5 which support the requirements to address the complete process of community detection. The tool is currently using the LiquidPub platform.

It is worth to note that part of the work done in this thesis was also presented and **accepted** in the conference *International Network for Social Network Analysis* (Sunbelt) to be held this year in April in Italy.

The problem of community discovery opens a wide field of different research lines that can be taken into account for future thesis. The current CET version supports the basic facilities needed for the discovery and analysis of communities, but there are many services and improvements that can be incorporated into the tool. Here are some research lines and services

that may be taken for future work.

1. **Communities Detection**

   The main component of the CET tool is the Clustering Engine (See Section 5.2). The incorporation of new algorithms for detection of communities in complex networks provide the user the ability to select the desired algorithm to discover communities. The main problem in these algorithms is the ability to handle large networks with low computational cost. The study of good heuristics based on maximizing modularity in the process of clustering can be part of a whole line of research for a thesis. The challenge of providing a name to communities is part of the detection process. This thesis took a first step by presenting a simple algorithm for name communities. However, this process requires a depth study of the information, using techniques such as text-mining, which contribute to get representative topics that can help the end user to identify the communities.

2. **Community Mining**

   The analysis of the discovered communities is important in order to understand their structure and provide services that can help to improve search and evaluation of researchers and scientific contributions. For example, one future work in this field is to help the user to better understand the reason for belonging to a particular community.

3. **Evolution of Communities**

   Communities change over time, the study of the behavior of communities over time is a whole line of investigation which is left for future work.

4. **Metrics**

   The metrics proposed in this thesis give rise to future analysis and subsequent research on evaluation metrics. The incorporation of new metrics to the tool is also part of future work.

5. **CET Services**

   The communities identified by the tool should be made available to other components of the LiquidPub platform. This requires the development of the **CET Service Module**

that works with extracted data from the tool and exports the necessary services for navigation, searching and analysis of communities. The module is in development and currently offers basic navigation and search, for which we detail as future work.

6. **Query UI**

   A small but important and pending task is the development of user interfaces (UI) that allow search different types of scientific entities in the tool.

As a final conclusion, this work has contributed significantly to the development of a new paradigm (proposed by LiquidPub) that takes the advantages of the WEB 2.0 and social networks in order to model a new mechanism of elaboration, evaluation and distribution of scientific content.

Appendix A

Community Engine Tool (CET)

A.1 Packages

The project source files are organized in a set of packages:

- **org.communityengine.cluster**: this package contains all the classes for the detection of communities.
- **org.communityengine.convert**: this package holds the classes for the conversion between models.
- **org.communityengine.io**: it contains classes that manage the input/output of the tool.
- **org.communityengine.manager**: the models are managed by the classes in this package.
- **org.communityengine.model**: it contains the model for each entity, such as author, conference, and so on.
- **org.communityengine.naming**: the naming process of the community are developed by classes in this package.

- **org.communityengine.network**: it contains the classes that builds the different networks needed by the tool.
- **org.communityengine.resource**: this package contains the classes to access different resources such as Reseval or databases.
- **org.communityengine.test**: test cases for classes.
- **org.communityengine.ui**: the package contain the classes for the user interface. All the different layouts are in this package.
- **org.communityengine.ui.decorator**: decorator classes for the ui package.
- **org.communityengine.util**: utility package.

A.2 Export Format of Communities

A.2.1 GCT Format

The GCT Format is used in order to load the communties found by the CET tool to another tool called Group Comparison. Group comparison is a tool, developed for the LiquidPub Project, that allow you to create groups of researchers and then evaluate and compare them using several metrics, like h-index, g-index, number of publications and citations, average number of citations, etc. You can create a group browsing universities, sectors, departments, faculties or simply adding your co-authors or your research team to your personal group. Once groups are created you can compare researchers within a particular group, comparing their h-index, g-index, number of citations or publications, etc. It is also possible to do comparison across groups, selecting two or more groups. You can compare global indexes among groups to discover the more productive ones or the highly cited ones.

The format is described as follow:

GCT Format

```
<group-list>
  <group>
    <feature>
      <name>THE NAME OF THE COMMUNITY</name>
      <description>TAGS</description>
    </feature>
    <author>
      <firstName></firstName>
      <middleName></middleName>
      <lastName></lastName>
    </author>
    <author>
      <firstName></firstName>
      <middleName></middleName>
      <lastName></lastName>
    </author>
  ...
   </group>
   ...
</group-list>
```

Table A.1: XML Format compatible with the GCT

A.2.2 CSV Format

In this format is exported the different scientific entities and the communities they belong.

1. $Author \times Community(AC)$: (GROUP_NUMBER, GROUP_NAME, AUTHOR_NAME, AUTHOR_PUBLICATIONS)

   An example is given in Table A.2 which correspond to group number 29, the two biggest conference in te community are WWW and ICDE, then the name of the author and the number of publications in the community.

2. $Conference \times Community(CC)$: (CONFERENCE, GROUP_NUMBER, GROUP_NAME)

   Table A.3 shows an example of the Conference Membership Format.

3. $Author \times Community - Extended(ACE)$: (GROUP_NUMBER, GROUP_NAME, AUTHOR, PUBLICATION_IN_THE_COMMUNITY, H_INDEX, G_INDEX, TOTAL_CITATIONS)

   This format export the same information as the AC format, with the extra information of some metrics obtained from Reseval[25] which are te h-index, g-index and total

CSV

```
29,[www- icde],Belle L. Tseng,10
29,[www- icde],Ben Adida,1
29,[www- icde],Ben Blum,1
29,[www- icde],Ben Bratu,1
29,[www- icde],Ben Calderhead,1
29,[www- icde],Ben Carterette,13
```

Table A.2: CSV Format for Author Membership

CSV

```
conf/sac/2008,2,[sac- compsac]
conf/seke/2008,2,[sac- compsac]
conf/icdim/2008,2,[sac- compsac]
conf/ispa/2008,2,[sac- compsac]
conf/apsec/2007,2,[sac- compsac]
conf/wetice/2008,2,[sac- compsac]
```

Table A.3: CSV Format for Conference Membership

citation count. See Table A.4.

CSV

```
conf/sac/2008,2,[sac- compsac]
conf/seke/2008,2,[sac- compsac]
conf/icdim/2008,2,[sac- compsac]
conf/ispa/2008,2,[sac- compsac]
conf/apsec/2007,2,[sac- compsac]
conf/wetice/2008,2,[sac- compsac]
```

Table A.4: CSV Format for Author Membership with metrics

A.3 Community ER Model

The discovered communities are hosted in a database in order to make available to other components the data obtained by the tool. Figure A.1 shows the Entity Relational (ER)

Model that was designed by the Liquid Pub team.

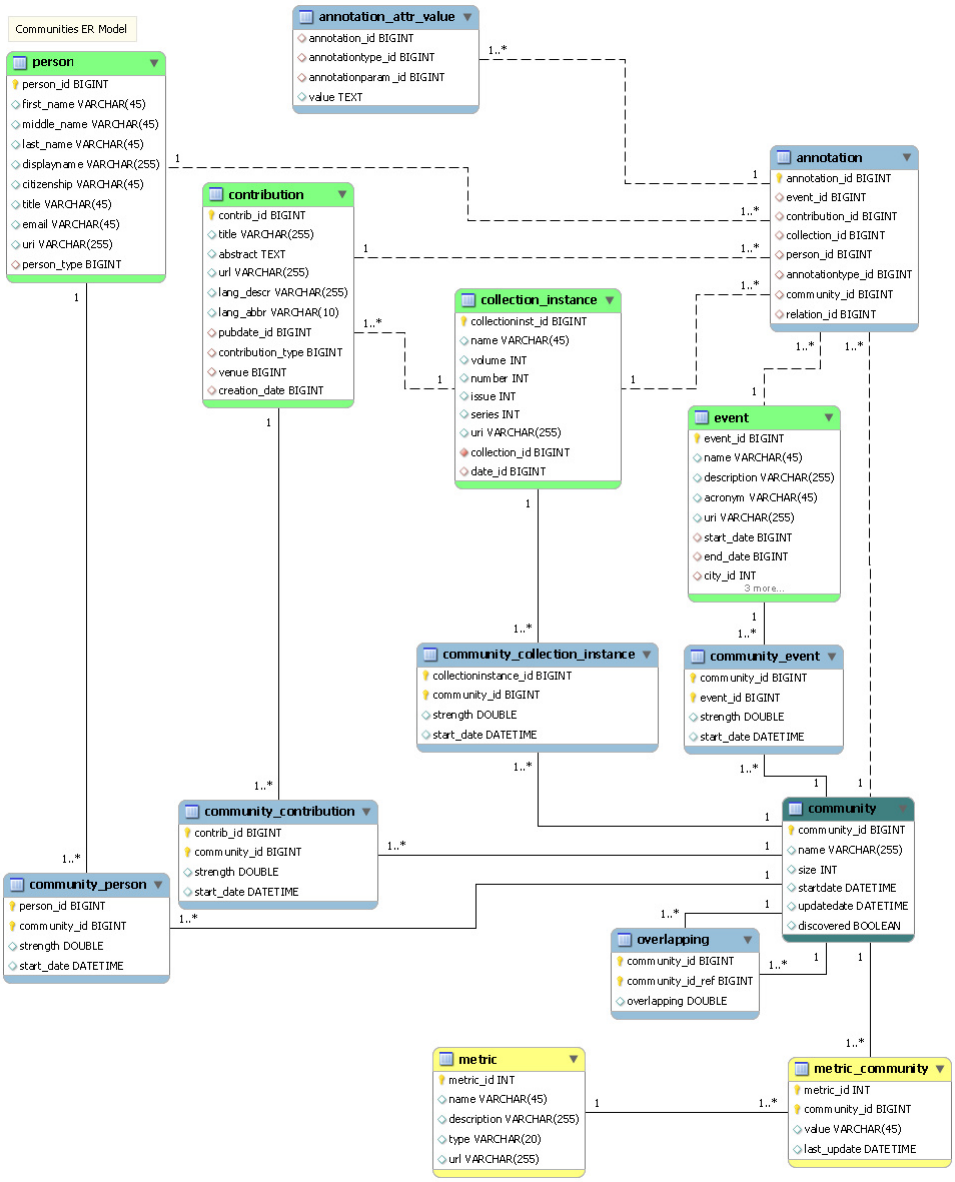

Figure A.1: Community ER Model

Appendix B

Additional Information of the analysis with ORA

The chapter describes the input format of ORA and the analysis made to different scientific networks.

B.1 DyNetML XML

The different networks being analyzed were represented into a DynetML format compatible with ORA. Table B.1 shows an example of the DynetXML format used to load affiliation network.

B.2 Authorship Network Analysis

Figure B.1 details the centrality analysis made to this network. A high value of centrality out degree in this network corresponds to an author with high publication number.

Dynet XML

```
<?xml version="1.0" standalone="yes"?>
<DynamicMetaNetwork id="Affiliation">
   <MetaNetwork id="Affiliation">
  <nodes>
   <nodeclass type="Organization" id="Organization">
    <node id="2,783"/>
    <node id="12,719"/>
   </nodeclass>
   <nodeclass type="Agent" id="Agent">
    <node id="6,545"/>
    <node id="7,220"/>
   </nodeclass>
  </nodes>
  <networks>
   <network sourceType="Agent" source="Agent" targetType="
        Organization" target="Organization" id="
        Affiliation">
    <link source="6,545" target="2,783"/>
    <link source="6,545" target="12,719"/>
    <link source="511,471" target="8,754"/>
   </network>
  </networks>
   </MetaNetwork>
</DynamicMetaNetwork>
```

Table B.1: DyNetXML Format compatible with ORA

B.3 Citation Network Analysis

Figure B.2 shows an analysis of the citation graph, and external/internal link analysis of CONCOR and Newman algorithm are detailed in Figure B.3 and B.4 respectively.

B.4 Complete Network

In Figure B.5 the sphere of influence of node *Fausto Giunchiglia* is shown. Yellow nodes represent scientific contribution entity, red nodes represent people, and green nodes represent affiliations.

Node-Level Measure	Avg	Stddev	Min/Max	Min/Max Node
Centrality, Out Degree	0.0049	0.0076	0.0006	140 nodes (29%) have this value
			0.0719	Luca Benini
Centrality, In Degree	0.0049	0.0018	0.0041	2630 nodes (83%) have this value
			0.0166	10 nodes (0%) have this value

Figure B.1: Node-Level Measure of Authorship Network

Network Level Measure			
Average Distance	1.5095	Link Count	1203
Breadth, Column	0.6603	Link Count, Lateral	0.9983
Breadth, Row	0.478	Link Count, Pooled	1
Clustering Coefficient, Watts-Strogatz	0.0631	Link Count, Reciprocal	0.0042
Communicative Need	0.0042	Link Count, Sequential	0
Component Count, Strong	1087	Link Count, Skip	0.1704
Component Count, Weak	176	Network Centralization, Betweenness	0.0001
Connectedness	0.1049	Network Centralization, Closeness	0
Count, Column	1092	Network Centralization, Column Degree	0.0218
Count, Node	1092	Network Centralization, In Degree	0.0218
Count, Row	1092	Network Centralization, Out Degree	0.0218
Density	0.001	Network Centralization, Row Degree	0.0218
Diameter	1092	Network Centralization, Total Degree	0.0136
Diffusion	0.0016	Network Levels	6
Efficiency	0.9954	Redundancy, Column	0.0014
Efficiency, Global	0.0185	Redundancy, Row	0.0016
Efficiency, Local	0.1353	Span Of Control	4.6092
Fragmentation	0.8951	Speed, Average	0.6625
Hierarchy	0.9974	Speed, Minimum	0.1667
Interdependence	0.0008	Transitivity	0.2912
Isolate Count	1	Upper Boundedness	0.0551

Figure B.2: Network-Level Measure

Group	Internal link count	External link count	% Internal links	Silo Index
1	767	122	86.28%	0.7255
2	314	122	72.02%	0.4404

Figure B.3: External/Internal Link Analysis of CONCOR Algorithm

Group	Internal link count	External link count	% Internal links	Silo Index
1	18	0	100%	1.0000
2	22	1	95.65%	0.9130
3	98	3	97.03%	0.9406
4	10	0	100%	1.0000
5	5	0	100%	1.0000
6	6	0	100%	1.0000
7	1	0	100%	1.0000
8	1	0	100%	1.0000
9	7	0	100%	1.0000
10	60	3	95.24%	0.9048
11	1	0	100%	1.0000
12	106	6	94.64%	0.8929

Figure B.4: External/Internal Link Analysis of Citation in Newman Algorithm

Figure B.5: Sphere of Influence of node "Fausto Giunchiglia"

Bibliography

[1] An Informatics Europe report. Research evaluation for computer science. Prepared by the Research Evaluation Committee of Informatics Europe. Version 6.0, 20 May 2008.

[2] Anthony, A., and DesJardins, M. Open problems in relational data clustering. In *Proceedings of the ICML Workshop on Open Problems in Stastistical Relational Learning* (2006), Citeseer.

[3] Association for Computing Machinery. The acm digital library. Website. `http://portal.acm.org/dl.cfm`.

[4] Automap. Automap. Website. `http://www.casos.cs.cmu.edu/projects/automap/`.

[5] Baird, L. M., and Oppenheim, C. Do citations matter? *Journal of Information Science 20*, 1 (1994), 2.

[6] Batista, P. D., Campiteli, M. G., and Kinouchi, O. Is it possible to compare researchers with different scientific interests? *Scientometrics 68*, 1 (2006), 179–189.

[7] Blondel, V., Guillaume, J., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment 2008* (2008), P10008.

[8] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks.

[9] Brandes, U., Delling, D., Gaertler, M., Goerke, R., Hoefer, M., Nikoloski, Z., and Wagner, D. Maximizing modularity is hard.

[10] Casati, F., Giunchiglia, F., and Marchese, M. Publish and perish: why the current publication and review model is killing research and wasting your money. *Ubiquity 8*, 3 (2007), 1 – 1.

[11] CASOS - ORA. ORA. Website. `http://www.casos.cs.cmu.edu/projects/ora/`.

[12] Egghe, L. An improvement of the h-index: the g-index. *ISSI Newsletter 2*, 1 (2006), 8–9.

[13] Gate. Gate. Website. `http://gate.ac.uk/`.

[14] Girvan, M., and Newman, M. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences 99*, 12 (2002), 7821.

[15] Google. Google scholar beta. Website. `http://scholar.google.com.py/`.

[16] Hirsch, J. E. An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences 102*, 46 (2005), 16569–16572.

[17] igraph. igraph. Website. `http://igraph.sourceforge.net/`.

[18] Jin, B. The AR-index: complementing the h-index. *ISSI Newsletter 3*, 1 (2007), 6.

[19] JUNG. Java Universal Network/Graph Framework. Website. `http://jung.sourceforge.net/`.

[20] Kornfeld, W. A., and Hewitt, C. E. The scientific community metaphor. *IEEE Transactions on Systems, Man, and Cybernetics 11*, 1 (1981), 24–33.

[21] Krapivin, M., and Marchese, M. Focused page rank in scientific papers ranking. In *Proceedings of the 11th International Conference on Asian Digital Libraries: Universal and Ubiquitous Access to Information* (2008), Springer, pp. 144–153.

[22] Lead partner: UNITN. Contributing partners: IIIA-CSIC, SPRINGER, CNRS. D5.1v1. design of the liquid publications integrated platform. Tech. rep., University of Trento, 2009.

[23] Leximancer. Leximancer. Website. `https://www.leximancer.com/`.

[24] Ley, M., and Reuther, P. Maintaining an online bibliographical database: The problem of data quality. *Extraction et gestion des connaissances (EGC2006), Actes des sixiemes journées Extraction et Gestion des Connaissances, Lille, France* (2006), 17–20.

[25] LiquidPub Project. ResEval: research impact evaluation tool. Website. `http://project.liquidpub.org/reseval`.

[26] LiquidPub Project. ResMan: the resource space management module for liquidpub. Website. `http://project.liquidpub.org/resman`.

[27] Mann, G. S., Mimno, D., and McCallum, A. Bibliometric impact measures leveraging topic analysis. In *Proceedings of the 6th ACM/IEEE-CS joint conference on Digital libraries* (2006), ACM New York, NY, USA, pp. 65–74.

[28] Neville, J., Adler, M., and Jensen, D. Clustering relational data using attribute and link information. In *Proceedings of the IJCAI Text Mining and Link Analysis Workshop* (2003), Citeseer.

[29] Newman, M. Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E 64*, 1 (2001), 16131.

[30] Newman, M. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E 64*, 1 (2001), 16132.

[31] Newman, M. Analysis of weighted networks. *Physical Review E 70*, 5 (2004), 56131.

[32] Newman, M. E. J. Analysis of weighted networks.

[33] Newman, M. E. J., and Girvan, M. Finding and evaluating community structure in networks. *Arxiv preprint cond-mat/0308217* (2003).

[34] Newman, M. E. J., and Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E 69*, 2 (Feb 2004), 026113.

[35] Oppenheim, C. Do citations count? Citation indexing and the Research Assessment Exercise (RAE). *Serials: The Journal for the Serials Community 9*, 2 (1996), 155–161.

[36] Parra, C. *Infraestructura web para comunidades y recursos cientificos.* Universidad Nacional de Asuncion-Paraguay, 2009.

[37] Rapid Miner. Rapid Miner. Website. `http://rapid-i.com/content/view/181/190/`.

[38] Ratprasartporn, N., Po, J., Cakmak, A., Bani-Ahmad, S., and Ozsoyoglu, G. Context-based literature digital collection search. *The VLDB Journal 18*, 1 (2009), 277–301.

[39] Schreiber, M. To share the fame in a fair way, hm modifies h for multi-authored manuscripts. *New Journal of Physics 10* (2008), 040201.

[40] Sidiropoulos, A., Katsaros, D., and Manolopoulos, Y. Generalized Hirsch h-index for disclosing latent facts in citation networks. *Scientometrics 72*, 2 (2007), 253–280.

[41] Tarma Software Research Pty Limitated. Harzing's publish or perish. Website. `http://www.harzing.com/pop.htm`.

[42] The Pennsylvania State University. Citeseer: Scientific literature digital library and search engine. Website. `http://citeseerx.ist.psu.edu/`.

[43] Thomson Reuters. ISI Web of Knowledge. Website. `http://isiknowledge.com/jcr`.

[44] Wang, X., Jiao, L., and Wu, J. Adjusting from disjoint to overlapping community detection of complex networks. *Physica A: Statistical Mechanics and its Applications 388*, 24 (2009), 5045–5056.

[45] Weka. Weka 3: Data Mining Software in Java. Website. `http://www.cs.waikato.ac.nz/ml/weka/`.

[46] White, H. D., and McCain, K. W. Visualizing a discipline: An author co-citation analysis of information science, 1972-1995. *JASIS 49*, 4 (1998), 327–355.

[47] Wu, F., and Huberman, B. Finding communities in linear time: a physics approach. *The European Physical Journal B - Condensed Matter 38*, 2 (March 2004), 331–338.

[48] Zhang, C. The e-index, complementing the h-index for excess citations. *PLoS One 4*, 5 (2009).